$2.00

kilobaud™
The Small Computer Magazine

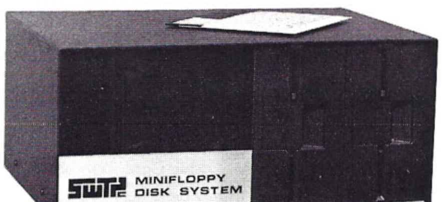
ISSUE #8 August 1977

Articles

Features

SWTPC announces first dual minifloppy kit under $1,000

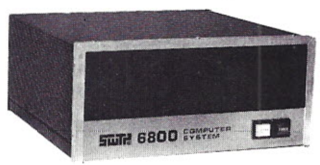
Now SWTPC offers complete best-buy computer system with $995 dual minifloppy, $500 video terminal/monitor, $395 4K computer.



$995 MF-68 Dual Minifloppy
You need dual drives to get full benefits from a minifloppy. So we waited to offer a floppy until we could give you a dependable dual system at the right price.

The MF-68 is a complete top-quality minifloppy for your SWTPC Computer. The kit has controller, chassis, cover, power supply, cables, assembly instructions, two highly reliable Shugart drives, and a diskette with the Floppy Disk Operating System (FDOS) and disk BASIC. (A floppy is no better than its operating system, and the MF-68 has one of the best available.) An optional $850 MF-6X kit expands the system to four drives.

$500 Terminal/Monitor
The CT-64 terminal kit offers these premium features: 64-character lines, upper/lower case letters, switchable control character printing, word highlighting, full cursor control, 110-1200 Baud serial interface, and many others. Separately the CT-64 is $325, the 12 MHz CT-VM monitor $175.



$395 4K 6800 Computer
The SWTPC 6800 comes complete with 4K memory, serial interface, power supply, chassis, famous Motorola MIKBUG® mini-operating system in read-only memory (ROM), and the most complete documentation with any computer kit. Our growing software library includes 4K and 8K BASIC (cassettes $4.95 and $9.95; paper tape $10.00 and $20.00). Extra memory, $100/4K or $250/8K.

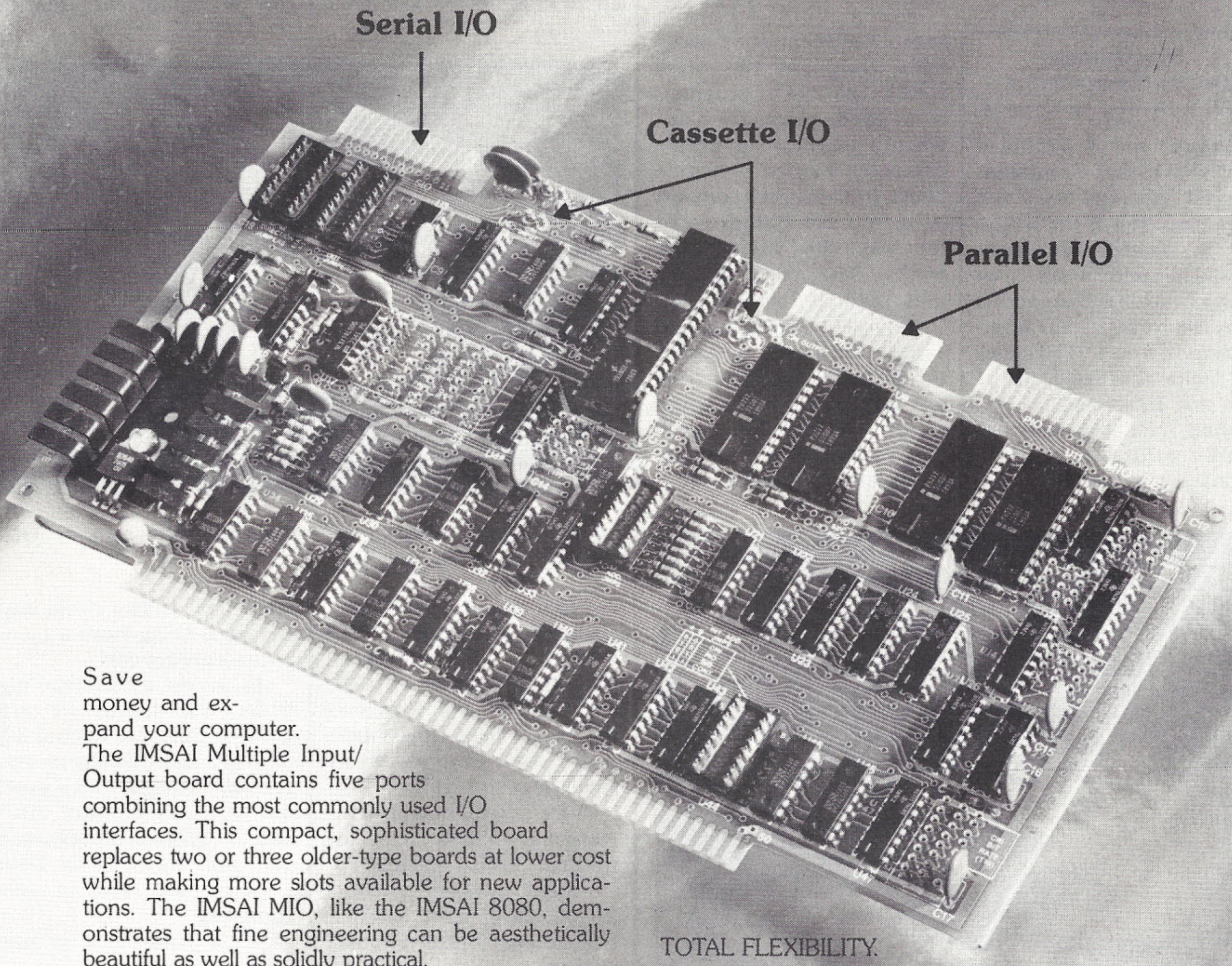
Other SWTPC peripherals include $250 PR-40 Alphanumeric Line Printer (40 characters/line, 5 x 7 dot matrix, 75 line/minute speed, compatible with our 6800 computer and MITS/IMSAI); $79.50 AC-30 Cassette Interface System (writes/reads Kansas City standard tapes, controls two recorders, usable with other computers); and other peripherals now and to come.

Enclosed is:

_____ $1,990 for the full system shown above (MF-68 Minifloppy, CT-64 Terminal with CT-VM Monitor).
_____ $995 for the Dual Minifloppy
_____ $325 for the CT-64 Terminal
_____ $175 for the CT-VM Monitor
_____ $395 for the 4K 6800 Computer
_____ $250 for the PR-40 Line Printer
_____ $79.50 for AC-30 Cassette Inferface
_____ Additional 4K memory boards at $100
_____ Additional 8K memory boards at $250
_____ Or BAC # __________ Exp. Date_____
_____ Or MC # __________ Exp. Date_____
Name_______________ Address_______________
City_______________ State _______ Zip_______

Southwest Technical Products Corp.
219 W. Rhapsody, San Antonio, Texas 78216
London: Southwest Technical Products Co., Ltd.
Tokyo: Southwest Technical Products Corp./Japan

IMSAI Presents:
The 3 for 1 Slot Saver
A Masterpiece in I/O Engineering.



Save money and expand your computer. The IMSAI Multiple Input/Output board contains five ports combining the most commonly used I/O interfaces. This compact, sophisticated board replaces two or three older-type boards at lower cost while making more slots available for new applications. The IMSAI MIO, like the IMSAI 8080, demonstrates that fine engineering can be aesthetically beautiful as well as solidly practical.

USEFUL AND EASY TO USE.

The IMSAI MIO is inexpensive and easy to install. It's the most advanced I/O board available, adding quality, convenience and value to your computer. With the IMSAI MIO you can control a keyboard, printer, 2 tape cassettes and a teletype/CRT simultaneously.

Consider these important "useability" features:

1. Easy to assemble components — all board identities are silk-screened.
2. Safe soldering — the entire board is solder masked for quick, fast assembly.
3. Fast, flexible configuration — options are selected with solderless wire strapping.
4. Easy testing and tuning — pre-recorded test routines on tape cassette.
5. Best of all — everything is explained in comprehensive, understandable documentation.

TOTAL FLEXIBILITY.

A control port, serial channel, 2 parallel I/O ports and a two-tape cassette interface provide all the versatility you need in an I/O board. The IMSAI MIO's flexible addressing and control features assure compatibility with non-IMSAI 8080 software. Tape cassette applications are enhanced through inclusion of the Byte/Lancaster and Tarbell recording modes.

ORDER NOW—SHIPPING NOW

The IMSAI MIO is available for immediate delivery. $195 in kit; $350 assembled. Write or call for more information. Specifications and feature product bulletin available on request. Send $1 for complete catalog of IMSAI products.

Prices: USA Domestic. Subject to change without notice.

IMSAI

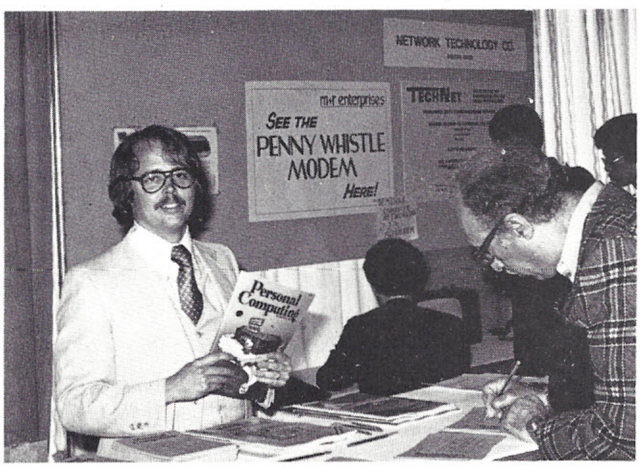
IMSAI Manufacturing Corporation
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093
TWX 910 366-7287

PUBLISHER'S REMARKS

Wayne Green

New Orleans Hamfest/Computerfest

Sherry and I attended the New Orleans Computerfest last year and we had a great time. It's at the Hilton Inn in Kenner this year, expanded quite a bit from last year. This is right across from the New Orleans International Airport. The dates are September 24-25th. If you are interested in exhibiting you can get full info by writing Box 10111, Jefferson LA 70181.



Dave Bunnell signs up a subscriber to his magazine, Personal Computing, *at his show in L.A.*

Big System Thinking

One of the yokes microcomputing has had to bear has been a carry-over of computer industry experience and thinking. The old-timers have learned some lessons and consider them immutable facts of life. Maybe, but in many respects the micro world is a whole new ball game.

I hate it when an author writes on and on in general terms . . . I like to consider arguments on the basis of specifics . . so let's consider some examples of differences between the maxi/mini world and the micro world. The sale of programs is a good place to start. When a business buys a big computer it hires a software house or some programmers to set up the system to fit the way the buying firm works. The programs are thus tailored to the buyer.

Anyone who has been in a small business knows that things don't work that way on that end of the spectrum. I envision small businesses buying a program for inventory and using it as it comes. The businessman will set up his inventory the way the program is written rather than the other way around. Ditto a payroll program, etc. Thus I can see a good market for well-written business programs which are adequately documented, rather than a market for programmers to write individual programs for a lot of small businesses.

Salesmen for the large systems patiently explained to me a couple of years ago that microprocessors would not significantly lower the cost of computer hardware because all they could cut was the CPU cost and we would still have to pay high prices for the other equipment. Baloney. We will see the costs on all peripherals and accessories coming down.

MOSTEK has come out with a dandy video generator chip . . . watch for articles on it in *Kilobaud* . . . four popular baud rates, character generator . . . the works . . . and it should shave $100 or more off the cost of a terminal. How long before we have floppy controller chips around? Oh, we have them already!

Michael Stone, Director of Marketing at Imsai, talks with me during a lull in activity.

The microcomputer market is all new . . . selling through stores instead of factory salesmen. Those $1000 equipment cabinets which large businesses buy as a matter of course would get laughed out of a computer store. Likewise a tab of $5,000 for a 32K memory board or $3,500 for an I/O board. There is no way for microcomputer hardware not to have a profound effect on big system hardware. Would you pay $1000 for an equipment cabinet from DEC if you could buy the same thing from a computer store for $95?

In a year or two the big firms will know we are around. Sure, we're cooking up 16K and 32K memories for our Altairs right now, but it won't be very difficult to make a few changes and put them in a Data General and save a few thousand.

With controllers getting

Computer Power and Light took up several booths for their Gene Murrow's Computer School. This innovation was very well-received and rank beginners were able in a few minutes to get a fair idea of what computers can do and the essentials of how they work.

cheaper, video terminals getting cheaper, printers getting cheaper . . . thinking will have to start changing. The lowering of prices is not just because of LSI development, it's also a result of enormous reductions in selling costs inherent in mass production.

Carole Ely shows off her new Vector Graphics system. Everyone was very impressed by the design and performance of this new system. You'll be reading a lot more about it as the word gets out.

Boggle

An old car rallying buddy called the other day. He'd wandered into a computer store in New York and found a copy of *Kilobaud* leaving with him. Later, on the subway, he jumped up in excitement . . . long lost Wayne Green had been found. I moved from New York to New Hampshire 15 years ago and haven't been back . . . very often.

It seems that my old friend Alan Turoff has gotten into inventing games . . . he'd send me a new one . . . he said, when he phoned me to renew our acquaintance. Sure enough, a couple days later a small box arrived . . . a Parker Brothers word game called Boggle. If you see one around, give it a try . . . you can learn the rules in one minute and start playing. It is very, very hard to stop.

Boggle is made up of 16 cubes with letters on them. You shake them up and then let them settle into a box with a four square grid to hold the cubes. You then have three minutes to make up as many three letter or more words as you can. Any number can play. When the time is up you eliminate any words found by two or more people and then score on the remaining words.

If you enjoy Scrabble, then you'll get a big kick out of Boggle. I like Scrabble enough so Sherry and I take a portable game with us to play on the plane

continued on page 13

Left: Nishi, the editor and publisher of Japanese I/O *magazine, tries out the new Southwest Tech TVT while designer Gary Kay looks on during the Los Angeles Personal Computing show.*

Right: Shelly Howard (R) of Micro Computer Devices had an IBM Selectric II, complete with interface for RS-232 which is going to sell for $1295. This should be quite a boon to people selling word processing systems . . . and to computerists who want first-rate copy from their computers. Hopefully more authors with word processing programs will buy these instead of dot matrix printers, which drive our typesetting department crazy.

Kilobaud is published monthly by 1001001, Inc., Peterborough NH 03458. Subscription rates in the U.S. and Canada are $15 for one year and $36 for three years. Outside the U.S. and Canada, please write for foreign rates. Application to mail at second class postage rate pending at Peterborough NH 03458 and additional mailing offices. Phone: 603-924-3873. INCLUDE OLD ADDRESS AND ZIP CODE WITH ADDRESS CHANGE NOTIFICATION.

EDITOR'S REMARKS

John Craig

Those Computers for the Masses

Why is there a *Kilobaud*? Well, it's because there are several tens of thousands of us "nuts" who are interested in computers as a hobby. Without a doubt, we're having a lot of fun but one of the most enjoyable aspects of this whole thing is the fact that we're all part of a pioneering effort which will culminate in millions of people owning home computers.

We're coming up to the point where this low-cost computer for the masses is right around the corner. Undoubtedly there will be several companies making a stab at producing this machine, but success will belong to those who come up with the right marketing formula. They can make the lowest-cost, best-looking machine with the most software, but ... if they don't take the right approach in convincing those millions of people they really *need* a home computer, it will all have been for nothing! They're going to need convincing, too. Probably nobody knows that better than we computer hobbyists because of our experiences in trying to relate to friends, relatives and strangers about how neat it is to have a home computer. We've seen those "so what" and "big deal" looks too often, haven't we?

It's going to take some big money. I say this because I firmly believe the medium to carry the message will have to be television ... and I mean prime-time television. Imagine a series of sixty-second commercials showing personal computers being used in the home for education, games, art, accounting, fire and security systems and in small businesses for payroll, accounts payable and receivable, and inventory. Boy, would I love to be the director who makes those commercials! It's really going to be a matter of having the right software packages so the machine can be effectively demonstrated as a useful tool. Those viewers aren't going to be the least bit interested in whether it's a 16-bit or 8-bit processor, whether it's constructed around the Altair or some other bus, whether it has DMA or vectored interrupt capabilities. No, they're going to be impressed by three things ... what it can do for them, how it looks, and how much it costs.

Commodore Business Machines' new PET just might be the one we've been waiting for. Briefly, the PET consists of a 6502-based processor, keyboard, cassette recorder and CRT built into a beautiful plastic molded case. PET is an acronym for Personal Electronic Transactor and I'd be willing to bet the reason they came up with that particular name is because it can be translated into other languages easily without loss of meaning. You see, the advertising brochure for the PET is printed in English, French and German. Kind of tells you something about their intentions, doesn't it?

The PET seems to have those three ingredients I mentioned earlier ... the looks, the software to convince people they need one and, most important, the price. It's going to be low, but not as low as the $495 they were originally shooting for. I don't want to get into the looks or the software because Sheila Clarke is preparing an article on the PET for next month's issue, and I'm sure she will cover those two aspects of the machine (along with many others).

Since Commodore was the company which came out with the first single-chip pocket calculator for under $200 back in 1970 and managed to revolutionize that industry ... it's quite possible they're going to be able to pull the whole thing off again. There's a big difference between selling a pocket calculator and something as sophisticated as a home computer system though. People didn't have to be convinced they needed pocket calculators, but they're going to need a selling job on the PET ... or any other home computer.

Commodore is taking a close look at marketing the PET through department stores, via direct mail, and through existing computer stores across the country. I'm glad to hear this because I think those stores and Commodore would do well to join forces. One of the ingredients for selling a product such as this successfully is that installation and after-sales service must be provided. Who could do this better than the computer stores scattered around the country? (The parallels between the growth of this industry and that of television in the late forties and fifties are going to be very similar ... watch.) Wayne organized an industry meeting in San Francisco last April (a day before the West Coast Computer Faire) and one result was a decision by attending dealers to form an association. Dick Heiser, owner of the Computer Store in Santa Monica CA, was particularly concerned about the stores being able to carry the PET and he feels an association is almost a necessity in this case. That famous slogan of the Three Musketeers would be very appropriate here but it should be slightly modified: "United we stand ... independently we fall."

The PET comes with an Operating System and BASIC in 12K of ROM *and* the system has been designed around the Hewlett Packard Interface Bus. I think we're going to see some interesting developments from these features. For example, we'll certainly be seeing an interface for the Altair to HP bus so that owners of Altair bus systems can take advantage of PET's low-cost peripherals. Or will it be the other way around? Maybe we'll see HP bus to Altair bus interfaces so PET owners can take advantage of the many peripherals and memory boards available? It'll probably be both.

Another interesting development will be a PET emulator built onto an Altair bus board. This will include the 6502 processor, 12K of ROM containing the OS and BASIC, and the PET cassette interface. Just wait ... with all that PET-developed software floating around in the years to come it will be difficult to ignore it even if we want to.

If you have any questions regarding the PET, Commodore would appreciate it if you would write (rather than call) for their brochures. Contact: Arnie Karush, Systems Sales Mgr., Commodore, 901 California Ave., Palo Alto CA 94304.

Well, one down and two to go ... the other two being Heath and Radio Shack. I don't know how much longer it'll be before we can tell you about what Radio Shack is doing, but you can look forward to a full report on the Heath system next month.

Articles

While I'm on this subject, here's a reminder to those writing for KB that articles should be sent to Peterborough rather than being sent directly to me in California.

Speaking of our writers ... we certainly do have the cream of the crop, don't we? *Kilobaud* is a great magazine and we shouldn't

continued on page 13

LEGAL/ BUSINESS FORUM

Kenneth S. Widelitz
Attorney At Law

So you're thinking of opening a computer store, marketing some software you've written, or going into a microcomputer manufacturing business with a friend. What are the business and legal considerations of doing so? Should you form a corporation or a partnership? How do you copyright your software?

Perhaps you want to form a local hobby computer club or, on a more grandiose scale, a nationwide nonprofit corporation to better society via microcomputers. How do you start? Can you get a tax exempt status? Possibly you've been ripped off by a mail-order house or have failed to receive satisfactory warranty service. What are the applicable laws? How do you obtain your remedies?

Have you wondered if Congress will legislate personal computing power (as the FCCC regulates personal communication power)? Are you familiar with the tax benefits of using your hobby computer in a business venture? Perhaps you're having trouble comprehending the theoretical model (i.e., that differential calculus equation for regression analysis used in predecting trends) on which a business applications program you're writing is based.

From the foregoing it should be apparent that the Kilobaud Legal/Business Forum is intended to run the gamut through all aspects of legal and business considerations and problems arising out of the hobby and business uses of microcomputers. The Forum may at times be practical, theoretical, philosophical, heretical, etc., ical.

The Forum will *not* take on a "Dear Abby" style. The California Rules of Professional Conduct state that: "A member of the State Bar shall not advise inquirers or render opinions to them through or in connection with a newspaper, radio, or other publicity medium of any kind in respect to their specific legal problems, whether or not such attorney shall be compensated for his services." Under no circumstances do I intend to run afoul of that rule or even come close. Therefore, I will not answer specific legal questions or even base discussions on a specific (although "hypothetical") state of facts.

I anticipate that the Forum will be a clearinghouse of information. Therefore I encourage correspondence describing legal or business problems that you have encountered and an outline of your solution, if any. I would be interested in hearing about litigation arising out of the use (or misuse) of microcomputers. I am especially interested in compatibility problems and intend to devote a future Forum to urge the formation of an industry-wide Informal Dispute Settlement Mechanism as envisioned by the Magnuson Moss Warranty Act as a method of dealing with the compatibility situation.

Your letters and comments will influence the direction the Forum takes, especially the topics discussed.

Federal Mail-order Legislation

One of the oft-heard complaints in the microcomputer industry is that "I ordered a product two months ago, sent in a check for the full amount of the purchase, and have heard nothing from the company, even though I've sent them two letters." Theoretically this situation should not arise. Unfortunately, many manufacturers, retailers, distributors, and most consumers are unaware that the Federal Trade Commission (FTC) has recently initiated the *Mail Order Merchandise Rule.* The Rule requires systematic correspondence between a seller (manufacturer, retailer, or distributor) and a buyer (you) where there is a delay in shipping merchandise after you have placed a prepaid order.

A failure to comply with the Rule makes the seller liable for a civil penalty of up to $10,000 for each violation. The penalty is enforced by lawsuits brought by the FTC.

The Rule requires that the mail-order house indicate in its solicitation (magazine ad, promotional literature) for an order the time in which merchandise will be shipped after receipt of the order. If no time is clearly and conspicuously stated in the solicitation, the merchandise must be shipped within thirty days of receipt of your order.

If a seller is unable to ship merchandise within the applicable time (time stated or thirty days), he must give you an option to either consent to a delay in shipping or to cancel your order and receive a prompt refund. The offer must be made to you within a reasonable time after the seller first becomes aware of his inability to ship your order within the applicable time; but in any event the offer must be made before the applicable time expires.

The offer must be made by the seller even if you have not inquired as to the status of your order. The offer must clearly and conspicuously disclose that you have a right to cancel the order and obtain a prompt refund. The offer should provide a definite revised shipping date if the seller has a reasonable basis to determine it.

When the seller provides you with a definite revised shipping date which is thirty days or less after the originally promised shipping date (thirty days or as originally stated) and you fail to respond to his offer, you are deemed to have consented to a delayed shipment.

If the definite revised shipping date is more than thirty days later than the original shipping date, or if the seller is unable to provide a definite revised shipping date, the rule is that your order will automatically be considered cancelled. That is, unless he ships the merchandise within thirty days of the original date and has received no cancellation prior to shipment (or he has received a response from you consenting to the shipping delay).

If there is an initial delay, the seller may request your express consent to a further unanticipated delay beyond the definite revised shipping date. However, even if you give express consent to further delays, you still have a continuing right to cancel your order at any time after the definite revised shipping date. You may do so by notifying the seller prior to actual shipment.

In the event of subsequent delays, the seller must provide you with a renewed option in which you may consent to a further delay or cancel your order and receive a prompt refund. The renewed option offer must be made within a reasonable time after the seller first becomes aware of his inability to ship before the definite revised date, but in no event later than the expiration of the definite revised shipping

continued on page 14

THE KIM FORUM

Rick Simpson
314 Second Ave.
Haddon Heights NJ 08035

In the last edition of the Forum, I promised to talk about the software that is now available for KIM and other 6502-based hardware. The question of software was probably raised more often than any other point by prospective KIM owners while I was managing KIM for MOS Technology.

The first request was usually for a good version of BASIC. One of the most popular is the version of Tiny BASIC offered for $5 in paper tape form by Tom Pittman of Itty Bitty Computers, PO Box 23189, San Jose CA 95153. Tom offers versions with their origin at either 400 or 2000(hex), depending on how your memory is organized. Tiny BASIC requires only 2K of memory and Tom provides a good user manual complete with notes on how to modify the I/O routines to meet your requirements. Like the OSI BASIC, no source listing is provided.

Last month I mentioned that the best source of KIM software was probably the *KIM/6502 User Group Newsletter*, edited by Eric Rehnke, 425 Meadow Lane, Seven Hills OH 44131. Issue 4 contains a novel game, KIMAZE, a relocation program for hand assemblers, a program to read that ID you forgot from your audio cassette, a chess clock, a cassette duplication program, and best of all, a calculator chip interface (with software) for the MOS 7529-103 IC. The interface uses the KIM I/O lines, the KIM, and the calculator chip – period. (Not the 29 ICs required in the recent calculator interface published in *Byte*) If you don't send $5 to Eric for the first six issues, you just aren't interested in 6502 software.

Also available for KIM is the best chess-playing program available for any microcomputer. Written by Peter Jennings, it's available for $10 from MicroChess, 1612-43 Thorncliffe Park Dr., Toronto, Canada M5H 1J4. Included are instructions on how to make it play a variety of opening games and modifications to vary its game strategy. An interesting feature is the ability to vary how long the program thinks about its next move – anywhere from 2 to 100 seconds. I had the pleasure of having dinner with Peter and some of his friends from Toronto at the Trenton Computer Festival. He mentioned that he had rewritten the Micro-Chess program to run on an 8080. The program requires all of the 1K memory on KIM, but requires more than 2K on the 8080!

Another excellent source of software is Bob Tripp of *The Computerist*, PO Box 3, S. Chelmsford MA 01824. Bob "retired" from his post with a well-known computer manufacturer to spend full time writing software and consulting in the micro field. His package runs on an unexpanded KIM-1 and allows you to play a variety of games including DAFFY (similar to Mastermind), Shooting Stars, and Hi-Lo. Also included are a digital clock and timer, a reaction timer and several other programs. All of his software is written in an intermediate language called PLEASE. Experienced programmers will recognize it as a series of functions with parameters which are passed to a small executive program.

Bob's latest effort, called HELP, is described as a cassette-oriented text editor which can be used to maintain mailing lists (finally, a *practical* application for your micro) and print selected portions as mailing labels. PLEASE and HELP are available from Bob for $10 each, complete with excellent documentation.

For TIM owners, a good collection of utility programs is available from The

continued on page 14

NEWS OF THE INDUSTRY

POLY 88 System Sixteen

A ready to run system, the POLY 88 System Sixteen lets you solve those home financing problems, perform a statistical analysis, or enjoy a host of challenging games. The 16K system features a high speed video display and an alphanumeric keyboard. Cassette tapes are used for permanent program storage. Programming is made simple by the BASIC software package.

PLOT and TIME are two of the unique features which rely on our video graphics and real-time-clock. Other features include Verify so that you know that your tape is good before you load another. Scientific functions, formatting options, and string capabilities are also included. In addition to the programs written by the user, the POLY 88 program library makes a growing number of applications available to the POLY 88 owner.

System Sixteen is priced at $2250 with kits starting at $735, and available from PolyMorphic Systems, 460 Ward Drive, Santa Barbara CA 93111.

Wire Dispenser Also Cuts and Strips

The new WD Series wire dispenser features unique cutting and stripping capability. Wire is drawn out of dispenser to required length. Then, a built-in plunger cuts length free from roll, while a gentle pull through the stripping blade removes the insulation without nicking the wire. Repeat procedure removes insulation from second end. Although designed particularly for wire-wrapping, the inexpensive dispenser is ideal for many applications. Dispenser includes 50 ft. (15m) roll of AWG 30 (0,25mm) top industrial quality Kynar® insulated OFHC silver plated solid copper wire. Insulation is offered in blue, white, yellow or red. Available from your local electronics distributor or directly from O.K. Machine and Tool Corporation, 3455 Conner Street, Bronx NY 10475.

System Sixteen by Polymorphic.

PCI Boards Offer Unlimited Potential

Both the Altair 88-Process Control Interface board and the new, similarly designed Altair 680b-PCI enable Altair computers to communicate with relays, switches, motors, fans, contacters, alarms, solenoids, lights, heaters and many other electromechanical devices. The 680b-PCI and the 88-PCI boards can be used in almost any instance where the computer must control large amounts of power.

Each board has eight relay outputs with SPST operation that are capable of switching 1 Amp at 120 V ac. But with the addition of relays, the amount of power that can be controlled is essentially unlimited. Both boards also have optically-isolated inputs, which can be configured to accept a wide range of input signals.

Two optically-isolated, software-controlled handshake lines are also provided for interfacing with external devices. All lines are isolated and balanced for operation in electrically noisy environments.

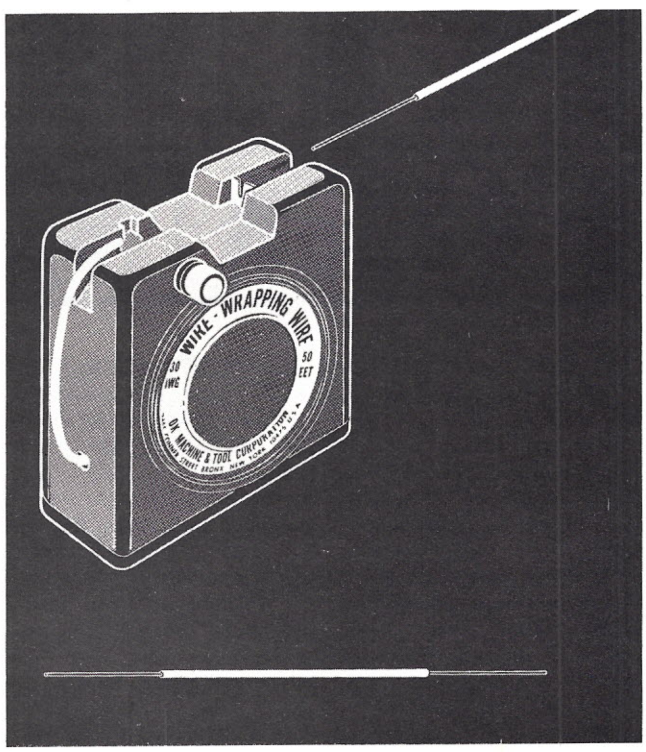
Each board is also equipped with a complete interrupt structure, which is under software control.

With the Process Control Interface Boards, the Altair computer can now be used in an even greater variety of applications. For example, by relaying sensory information to the computer, the PCI board allows control of lawn sprinklers, lights or thermostats. When used with such warning devices as trip wires, high temperature sensors or digital cameras, the board also allows the Altair computer to function as an alarm system. Mits, Inc., 2450 Alamo S.E., Albuquerque NM 87106.

Electronic Circuits Course

Heath Company, Benton Harbor, Michigan, has introduced a new learn-at-home electronics course covering

continued on page 15



O.K. Wire Dispenser.

BOOKS

Microcomputer Applications Handbook
David J. Guzeman
Iasis, Inc.,
Sunnyvale CA
1976, $7.95

If you open this book and expect applications, you're in for a big disappointment! Only one chapter is devoted to applications, and it consists of only two and one half pages. In that limited space five applications are suggested, with little attempt to indicate how one would proceed with the design. Even a block diagram as a minimum would have been appropriate for a book titled *Microcomputer Applications Handbook*.

The book consists of 8 chapters and 3 appendices. Six pages of Iasis promotional literature is thrown in for good measure.

For the hobbyist only chapters 5 and 6 are of any value. For the OEM (original equipment manufacturer) chapters 2 through 4 might be of interest. Chapter 1 would be helpful to those wanting an introduction on writing not-so-subtle sales literature. The chapter provides a negative discussion of evaluation boards and most presently existing microcomputer systems.

Chapter 5 is essentially the ia7301 three-ring binder microcomputer operational handbook. The chapter provides a schematic level discussion of the ia7301 circuitry. The level of writing and explanation here is excellent. Those hung up on cassette interfaces will find the Iasis approach interesting. It consists of one latch, one NAND gate, four inverters, a transistor, some LEDs, and a few capacitors and resistors.

Software appears in chapter 6, and is mainly concerned with writing the monitor program for the Iasis microcomputer described in the previous chapter. Flow diagrams and source listings are included. This is probably the best chapter of the book. Explanations are clear and to the point.

Chapter 7 is sales literature!

Appendix A is a brief introduction to the hexadecimal numbering system, while appendices B and C cover the 8080 architecture and instruction set.

Recommendations for this book: Borrow a copy and read chapters 5 and 6. These two chapters are well-written; unfortunately, they are not enough to carry the dead weight of most of the other chapters.

Bill Fuller
Grand Prairie TX

Some Common BASIC Programs
Lon Poole and Mary Borchers
Adam Osborne & Associates, Inc.
$7.50, paperback,
196 pages.

This book consists of listings of 76 different short to medium length BASIC programs. Each program listing is prefaced by a few sentences of introduction, and by copies of the output from a few sample runs of the program. Some of the programs are followed by "options," i.e., BASIC statements which may be used to make a slight alteration to the operation of the main program.

While there is no clear structure to the collection, the programs fall roughly into three categories: money matters, mathematics, and statistics. Here are a few examples from each category:

Money matters: nominal interest rate on investments, effective interest rate on investments, depreciation rate, depreciation amount, actual annual rate on a loan, mortgage amortization table, tax depreciation schedule.

Mathematics: linear interpolation, numerical integration, plot of polar equation, finding roots of equations, solving simultaneous equations, matrix operations.

Statistics: mean, variance, standard deviation, chi-square test, t-test, correlation coefficient, regression.

Other: linear programming (simplex method), day of the week (given date), converting English measures to metric.

The programs range from about 15 lines up to around 130 lines of BASIC. The average length is just over 37 lines (computed using their program for finding mean values). The amount of memory required by the programs will, of course, vary slightly depending on your specific version of BASIC. Using SWTPC's 8K BASIC, I've found that the programs in this book which don't use arrays take a little under 20_{10} bytes per line, including space for variables. In other words, you'll be able to run many if not most of the programs even if your system is a little short on memory.

The authors feel that even those with no prior experience with BASIC will be able to use the programs. This is probably so in many cases – since the program listings are printed directly from computer output (and are very legible as dot-matrix printing goes), there are probably few if any typographical errors in the listings. Thus, a person who is patient enough to keep checking against the book when faced with an error message will probably be able to get the programs going. Except for one small thing. The authors have chosen to use the non-standard assignment statement (i.e., they've left out the LET in assignment statements). The error messages that result on many versions of BASIC may prove obscure to the rank beginner. A very few of the programs require versions of BASIC which implement strings variables and arrays of strings.

Although people with virtually no knowledge of BASIC will be able to use the programs, I suspect that people with little knowledge of the subject matter of specific programs will have trouble seeing how to use the programs. I, for example, am woefully inadequate when it comes to accounting and money matters. As I read the description of the program entitled "discount commercial paper," for example, no light bulb flashes over my head . . . I have no idea what it means. Certainly people with no background in statistics will be unable to decide whether to use a chi-square or a t-test from the descriptions given with the programs. I mention this not as a criticism of the book but as a warning to the potential purchaser not to expect the impossible. If the book included substantial background material on each of the techniques used in the programs, it would swell to three or four hundred pages.

Some Common BASIC Programs is a welcome addition to the growing body of software aimed at people who know what they want to do with their microcomputer but don't know enough about programming to do it themselves.

Rich Didday
Santa Cruz CA

THE BASIC FORUM

Dick Whipple – John Arnold

In this month's BASIC Forum, we will attempt to answer questions about BASIC submitted by some of our readers. The first comes from Mickey Ferguson, PO Box 708, Trenton GA 30752. Mickey begins his letter with a critical comment directed squarely at BASIC Forum. Apparently he is concerned about our exclusive use of Altair BASIC in the first few BASIC Forum articles. While Altair BASIC was in the spotlight to start with, it was never our intention to write the "Altair BASIC Forum." Our initial use of this particular version of the language was prompted by our attempt to answer George Haller's question regarding a statement in the *Altair Basic User's Manual* (see the February issue of *Kilobaud*). In this and future BASIC Forums we want to extend the range of discussion to other BASICs. As in the past however, we will rely heavily on reader contributions to determine the main topics of discussion.

Now to Mickey's specific questions: Of what interest are statements like

```
WAIT6,1,1
OUT7,T(I)
LET T(I)=INP(7)
```

Here again, he doubtless refers to the BASIC Forum mentioned above. The WAIT, OUT, IN and, in addition, the USR, PEEK, and POKE statements of Altair BASIC were created to provide a way to link the BASIC program directly to certain machine level functions. As a programmer gains experience and writes more elaborate programs, he generally discovers there are some tasks that simply cannot be done conveniently (if at all) within the confines of the BASIC language he is using. Often it is possible to perform the task with the use of some machine level functions. These linkage statements provide a way for the programmer to "get at" machine functions from a BASIC program. Perhaps a few examples from our own experience will help.

1. The storage and retrieval of data on cassette tape as described in the March BASIC Forum is a good use of the WAIT, INP, and OUT. Without these statements, direct access to I/O devices other than the keyboard/printer would be very difficult.

2. We once needed a way to clear our CRT screen and home-up. While a BASIC program using the PRINT statement to issue blanks would work, it was not fast enough. (We were doing animation and needed a high speed clear screen function.) To circumvent the difficulty, we wrote a machine language routine that would clear the screen at full processor speed. The USR function linked to the machine language program, cleared the screen, and then returned to the BASIC program at the next statement.

3. A business we know utilizes the INP function to transfer data from an analog/digital converter to the BASIC program. The data enters the BASIC program in a decimal form that can be stored and used in whatever way the application requires. Similarly, the OUT statement could be used with a digital/analog converter to control an external device from the BASIC program.

We might add that these machine language statements are available on most 8K or larger BASICs currently available at the hobby level. The fact that they are not available on some larger commercial systems must be viewed as an innovative stroke on behalf of personal computing. Bill Gates and his associates at Micro Soft deserve much credit for originating those statements in Altair BASIC.

Next question: Can you explain the value of user-defined functions? As I understand it, I can say [Example 1], but why not just put the expression in wherever it is needed or perhaps put it into a GOSUB-RETURN?

```
10 DEF FNX(X)=1/(2*3.1415926)*F*C
```

Example 1.

Evaluate: $$C=A\cdot\left(\frac{\sin(w\cdot t)}{wt}+1\right)+B\cdot\left(\frac{\sin(2\cdot w\cdot t)}{2wt}+1\right)$$

GOSUB-RETURN Method

```
10 LET TEMP=W*T
20 GOSUB 100
30 LET C=A*TEMP
40 LET TEMP=2*W*T
50 GOSUB 100
60 LET C=C+B*TEMP
70 PRINT C
80 STOP
.
.
.
100 TEMP=SIN(TEMP)/TEMP+1
110 RETURN
```

User-Defined Function Method

```
10 DEF FNX(X)=SIN(X)/X+1
20 LET C=A*FNX(W*T)+B*FNX(2*W*T)
30 PRINT C
40 STOP
```

Example 2.

The principal advantage of the user-defined function is to save space in program memory. In most microprocessor BASICs, each ASCII character of the expression would require one byte of 8-bit wide memory. The expression above would require a total of 21 bytes. Suppose the expression appeared five times in the program. The total byte count would then be 105. With the expression set up as a user-defined function, then it can be referred to throughout the program by a short name containing only 5 or 6 characters. With a user-defined function for the expression above, memory use can be trimmed by 50 or more bytes. Of course the more references to the expression, the more memory saving there would be.

Using a GOSUB-RETURN would be satisfactory as long as the expression alone was to be evaluated. The great advantage of user-defined functions comes from the fact that they can appear as part of another expression. Trying to use a GOSUB-RETURN in such a situation would be awkward at best. Consider Example 2. The code efficiency of the User-Defined Function Method is clearly evident.

Now to a series of questions from another reader, Verlynn Johnson, RFD 2, Storm Lake IA 50588. Verlynn is relatively new to personal computing, although he has had programming experience on the big machines. He owns an 18K

continued on page 15

LETTERS

Recorders On! Recorders Off!

I've got to complain about a minor "bug" in *Kilobaud* #5. It is in the article titled "Make your investment count" by Phil Hughes. In this article Mr. Hughes states, "Also, 8K BASIC will not start and stop the cassette recorders under program control." This statement is not only misleading, it is completely incorrect! My wife has several programs (usable in both 4K and 8K BASIC) that do just this for the purpose of saving and inputting data instead of using very large matrices. The CHR$(X) function is used for this purpose.

PRINT CHR$(17) - - - turns the recorder on (to read a tape)

PRINT CHR$(18) - - - turns the recorder on (to make a tape)

PRINT CHR$(19) - - - turns the recorder off (stop reading tape)

PRINT CHR$(20) - - - turns the recorder off (stop making tape)

To put the variables A, B, and C on a data tape, the following would be used:

```
10 PRINT CHR$(18)
20 FOR X=1 TO 100
30 NEXT X
40 PRINT A,B,C;CHR$(20)
```

The FOR loop is used as a time delay to allow the recorder to get up to speed. Perhaps we are like the bumblebee who does not know that it is not possible for him to fly, so he flies anyway? We did not know that we could not turn the recorders on/off under program control when using SWTPC's 4K and 8K BASICs, so we have been doing it anyway!

Mickey Ferguson
Trenton GA

You're right. The SWTP 8K BASIC does have provisions for these commands. The "problem" (and error) lies in the fact that Phil's terminal (Lear Siegler ADM-3) doesn't provide circuitry for decoding such commands. (You do have this feature in your SWTP CT-1024, Mickey.) Next month we'll be running an article by Phil which discusses a solution to this problem. – John.

Handwritten Letters

A comment of yours in the "Editor's Remarks" section of Issue no. 6 perturbs me. You wrote, "It's also very sad to get really interesting letters which are handwritten! We can't retype those letters . . . no matter how good they are. And, we can't hand them to the typesetters if they're handwritten (period)."

My reaction: 1) Like, why can't you type up those really interesting letters? Aren't you really saying you won't take the trouble? 2) Or, why can't you give handwritten material to the typesetters? Who runs your act, anyhow? 3) Your intransigence seems particularly inappropriate coming just after your comments on the failure of certain businesses to provide enough *service.* Do you feel that service is only something received, never offered?

Mac Oglesby
Putney VT

I wasn't trying to get anyone perturbed, Mac . . . just making a feeble attempt at making things go a little smoother. (By the way, folks, Mac's letter was handwritten! But, it was very legible, and was actually handprinted *using upper and lower case letters.) I'll take your comments by the numbers, Mac:*

1. On occasion I have typed up really interesting letters, but that was in the early days before I had a million manuscripts staring me in the face!

2. Try reading some samples of handwriting being put out by people today. (And as far as who is running the act . . . I thought it was a well-known fact that the typesetters ran this whole operation! You should meet those three girls sometime!)

3. I never really looked at the "Letters" section as a service. *I've always thought of it as a media for sharing ideas and thoughts. I would appreciate it if those ideas and thoughts could be typed or neatly handprinted in upper and lower case letters, okay? – John.*

Regarding Math, The Doctor Says . . .

Re the "math or not to math controversy":

I consider myself well-credentialed to enter the ring as an expert for the following reasons: 1. My mother taught high school math for 50 years and was a theoretical mathematician. 2. My sister won a college scholarship based mainly on her math and statistics ability. 3. I can add 2+2 easily, but usually incorrectly. The following list of sweeping generalizations is the way I profile the two types:

Math people always
are of slender build
die of brain tumors
are picky eaters
win scholarships
analyze gambling odds
dislike "sales" psychology
are compulsive-obsessive
like Baroque and early classical music
are poor
are unable to repair bent safety pins
feel very comfortable at a keyboard
write for *Byte*
are interested in neither sex
like math games
design software that astounds (if plugged into real-world interfaces)

Non-math people always
are of athletic build
die from football injuries and bullet wounds
like MacDonald's cheeseburgers
win friends
gamble
become salesmen
are lazy
like top 40 pop hits
are well-to-do
are able to repair cars, radios, kitchen sinks, etc.
feel very comfortable with a soldering gun in hand
write for *Kilobaud*
are interested in sex
like parlor games
design real-world interfaces that are truly effective (if programmed effectively).

William J. Schenker, M.D.
Walnut Creek CA

Mail Chauvinism Raises It's Ugly Head Again!

When are you ham types going to grow up? How would you like to be referred to as an XYM? I find the use of the term XYL in "Kilobaud Klassroom, Part I," to be both demeaning and condescending. It seems to relagate the wife to the status of a piece of test equipment, like a TRVM or CRT on the shelf. The computer hobby is a new one, and the assumption that your audience is exclusively male should not be made. It only discourages women from becoming interested. We have money and buy subscriptions, too; I just bought one!

I've always been peeved by the fashion-plate model at the console of the IBM computer, because, as a computer-programmer, I know how few women are ever hired as operators. I'm also very peeved with Amp'l Anny in the Tri-Tek ads. I'm peeved with her for two reasons, first, it assumes a completely-male buying audience, and second, she projects the "big-boobs-no-brains" image.

I think you owe an apology to the ladies in your audience.

Leah R. O'Connor
Chicago IL

First, I'd like to say thank you for the nice letter, Leah (whew!). Secondly, I'd like to say that if I ever feel I owe an apology to the women readers of Kilobaud *(or any other group) I'll do it. I don't feel one is due now, though.*

George Young is a ham. For years and years hams have been calling their wives Ex-Young Ladies (XYLs) and it's an endearing term . . . certainly not intended to be demeaning or condescending. I personally prefer YL and you'll find a large number of hams feel the same way. You realize of course that the hams wives use OM (Old Man) as an endearing term for their husbands.

If I were you I'd con-

continued on page 108

PUBLISHER'S REMARKS

from page 3

while flying to conventions. We also take a cribbage set along and a Master Mind. These things make plane trips much more bearable.

Should you get Boggled, let me know.

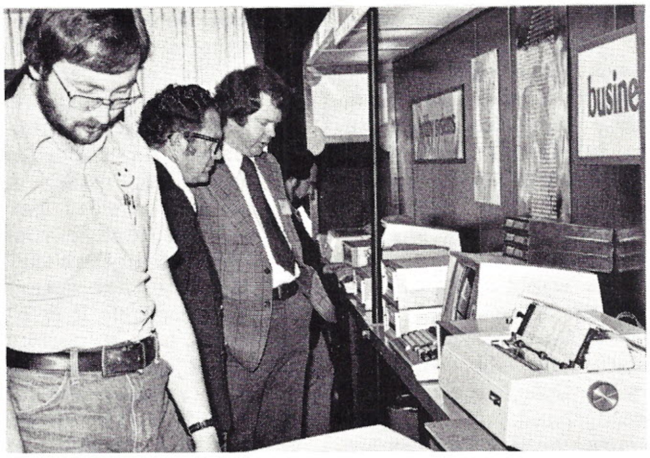
Call for Papers

An extensive symposium on hobby and small business computing is being scheduled for COMPUTERMANIA, August 25-27th, in Boston.

Papers are hereby solicited for presentation for possible publication in the Program Magazine. This will not prohibit further publication of the paper in *Kilobaud*, etc., or the payment of regular article rates for such publication.

Papers should be typed double-spaced and be well-illustrated.

The deadline for all papers for COMPUTERMANIA is August 1, 1977.



The Mits booth was kept busy at the Los Angeles show.

Papers are solicited from both hobbyists and industry. Papers will be accepted on the basis of interest value to microcomputer users.

Please send papers to COMPUTERMANIA, c/o *Kilobaud*, Peterborough NH 03458.

Big Daddy Put-Downs

Twice recently I've seen computer hobbyist put-downs in the professional computer magazines . . . snide comments about hobbyists managing to repeat the mistakes made by big machine programmers 20 years ago. Possibly. Wouldn't it be nice if these old-timers would stop laughing at our mistakes long enough to write some articles which would help lead us through the woods.

Our senior citizen programmers would do well to try our cute little microcomputers and get a hint of what is coming . . . and perhaps join in the fun.

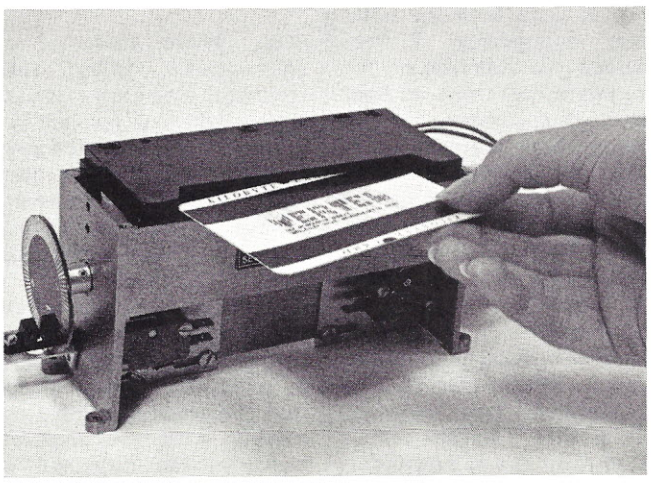
Enhancing the Fairchild

The Fairchild video game unit sells for $169 in the stores and comes with some cartridges of game programs. A reader suggests that someone get busy and connect a keyboard to the setup. Why not? I'd like to see some info in *Kilobaud* on the Fairchild system and data on I/O to it from our microcomputers, keyboards, etc.

Vertel Kilobyte Cards

There has been a growing need for a low-cost, easy-to-use, sturdy programming medium which would work with home video games. Vertel has come up with a good answer to the problem: a credit card with magnetic stripes on it which will hold up to one kilobyte of data. The machinery for making and handling credit cards is popular and inexpensive, so it is a good physical medium. Credit cards are easy to handle, store and use. Vertel has a unit which records the data on the cards and/or reads the data . . . plus the I/O board and controller . . . all in the hobby price range.

Compared to the Fairchild cartridges, cassettes and other popular programming methods, the Vertel system seems to have much to offer.



The Vertel Kilobyte card.

EDITOR'S REMARKS

from page 4

forget for a moment that all the fine writers are responsible for its success. Some day I'd like to throw a big party for every one of them!

Now . . . on to the topic of the day . . . articles. The time has come for us to start seeing more sophistication in our games. I would much rather see games which challenge the player's intelligence, agility and imagination than those which serve only as a substitute for a pair of dice. Those using a computer as simply a random number generator should be prosecuted for wasting their computer's resources!

I'm constantly amazed at how difficult it is to pry an article out of some manufacturers about their products. If I were going after this hobbyist market with a board, a piece of software, or a complete system I would make absolutely sure my marketing effort included an article describing the applications in which it could be used (i.e., make the reader *want* it), the design considera-

tions and trade-offs and the theory of operation. The article would contain photos, diagrams, listings, flowcharts and (in the case of a board) complete construction details which would include the PC board layout and artwork. (That last one might unsettle the poor guy who feels this would be the same as giving his product away . . . but there's really no cause for alarm. Sure, there are going to be a few people who will actually make a negative and make their own PC board . . .but the number will be incredibly small compared to those who purchase the board.)

Of course, there are those who don't really need the exposure a good article would give their product. Take Bob Mullen (if only someone would!). Mullen Computer Boards recently completed development of an opto-isolator/relay control board for the Altair bus. The endless applications for this board are mind-boggling! But, unfortunately we'll probably never see an article on it because Bob is too tied up with taking cruises on his yacht and counting his money! Like I said, there are some out there who don't need the exposure. (It's important that the word "exposure" be emphasized here because articles written by manufacturers about their products *cannot* come off sounding like advertisements.)

If you really can't find the time to write about your product, or for some other reason you can't (or won't) do it, then drop me a line and let's see about getting it into the hands of one of KB's writers for a hardware or software review. There's a lot to be said for an objective review.

Program Listings

I'm sure a lot of you noticed the program listings in last month's issue which were not typeset. Barbie, Sandy and Marie (our typesetters) undoubtedly got together and threw a party to celebrate this decision. Aside from relieving those girls of some tedious work, there are other advantages to be derived from using computer-generated listings . . . the most important being the reduction of possible errors.

We would prefer that *all* articles which contain programs have the listings generated by the computer rather than being typed or handprinted. If the programs are *lengthy* then we (and the readers, of course) would be most grateful if every attempt was made by the author to generate those listings in camera-ready form. This can usually be accomplished by putting a new ribbon on your printer or TTY before running off the listing.

The KIM Forum

from page 7

6502 Program Exchange, 2920 Moana, Reno, NV 89509. Write for their latest list.

While I was with MOS Technology I was contacted by a programming club at the Colorado School of Mines who had developed some excellent large-scale software and was willing to share both source and object code with the 6502 user community. I have agreed to distribute this software for them so that they can continue to develop more software without the hassles of duplicating listings and tapes and interfacing with users. As of now there are three packages available; a 4K version of FOCAL, a 2-1/2K resident assembler, and a high-level language compiler called XPLO, FOCAL is a registered trademark of Digital Equipment Company for a BASIC-like language and includes floating-point variables, one-dimensional arrays, user formatting and a complete editor. The assembler uses the MOS Technology op codes but a slightly different scheme for indicating the addressing mode. A loader is also provided. You will have to patch in your own I/O routines since they will depend on the devices you have available. I don't have room here to go into details but an information packet is available for $2 from ARESCO, 314 Second Ave., Haddon Heights NJ 08035. The complete packages, including source listings (over 100 pages for FOCAL), user manuals, and object code on cassette or paper tapes are available from the same address. FOCAL is $40, the assembler/loader is $30 and XPLO is $40.

As a final note for the hams in the audience, I'm told there is a RTTY autostart net on 3637.5 kHz which includes some KIM-1 users. I've always thought this would be a great way to distribute software. Now can someone figure out how to copy RTTY with a KIM? The on-board PLL should work fine as an AFSK demodulator!

Legal/Business Forum

from page 6

date. Where there is a subsequent delay, the renewed option must inform you that you will be deemed to have rejected any further delay unless the seller receives a consent from you prior to the old definite revised shipping date.

The failure by the seller to provide the required notices and options creates a rebuttable presumption that the seller has failed to comply with the requirements of the Rule and is liable for the civil penalty. A rebuttable presumption is a rule of evidence which says, in essence, that a certain "fact" is presumed to be true. However, it is possible to rebut the presumption and show that the presumed "fact" is not actually true. It is important to note that the same rebuttable presumption is created if the seller fails to provide you with the means, postage prepaid, to notify him regarding your decision with respect to the option.

The refund, in order to be prompt, must be sent to you by first class mail within seven working days of when the seller receives notification from you that you wish to cancel your order. In a situation where your order is considered cancelled because of an initial delay of more than thirty days (or an inability to provide a definite revised shipping date), the refund must be made within one billing cycle of the seller.

The failure to make a prompt refund also subjects the seller to a civil liability of up to $10,000 per violation.

If you feel that a company you have been dealing with is in violation of the Mail Order Merchandise Rule, you should file a complaint with the FTC. The FTC has complaint forms and will provide you with one following a telephone call or written inquiry.

Not all complaints are investigated by the FTC. The management at the FTC office first evaluates the complaint to determine whether it is in the public interest for the FTC to conduct an investigation. The factors considered in determining whether or not an investigation is in the public interest are primarily the size of the non-complying company and the extent of the consumer injury. The extent of the consumer injury is ordinarily the amount of the purchase. Purchases of microcomputer products will generally be of a sufficient size to be worthy of an investigation. If the investigation turns up evidence which indicates a

violation, the FTC will file a complaint against the offending company.

As a practical matter, perhaps the best way to approach a situation where you have not received merchandise you have ordered within the promised period (or, if no promised period, within thirty days), would be to write a letter to the FTC, with a copy to the offending company, describing the nature of the problem and requesting a complaint form. At least that will tell the offending company that you mean business.

Kilobaud Legal/Business Forum
10960 Wilshire Blvd.
Suite 1504
Los Angeles CA 90024

NEWS OF THE INDUSTRY

from page 8

basic electronic circuits. The course, EE-3104, is one of four basic electronics courses which use programmed instructions plus audio records. The course comes complete with electronic parts for "hands on" experiments. Other courses in the basic electronics series include AC Electronics, DC Electronics, and Semiconductor Devices. An advanced course in Digital Techniques is also available.

Course EE-3104 covers basic and operational amplifiers, power supplies, oscillators, pulse circuits, modulation and demodulation with emphasis on integrated circuits. An optional final exam can be taken for Continuing Education Units (CEUs), a nationally recognized means of acknowledging participation in non-credit adult education.

Courses are mail-order priced at $39.95. For further information, write for a free catalog to: Heath Company, Dept. 350-18, Benton Harbor MI 49022.

CT-6 Terminal System

The Southwest Technical Products Corporation CT-64 Terminal System kit along with the optional CT-VM video monitor is a complete package providing everything needed for a complete stand-alone terminal system compatible with modems and ASCII computer systems of every kind.

The kit features 16 lines of 32 or 64 characters per line, scrolling or page mode operation, upper and lower case characters, reversed character printing, control character printing, cursor control and complete control character decoding.

The kit includes the power supply, keyboard, serial interface, beeper, assembly instructions, chassis and cover and is sold in kit form only for $325 ppd in US. The optional CT-VM video monitor. is sold assembled and requires the CT-64's power supply. It sells for $175 ppd in US from Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio TX 78216.

OSI 460Z CPU Expander

The OSI 460Z CPU Expander's main purpose is to allow a user to run 8080, Z-80, and 6100 (PDP-8) software on his 400 system without modifying the software.

But there is much more. The 460Z is inserted in the 400 bus between a 6502 based 400 board with optional system boards and the rest of the 400 system. The 460Z contains a Z-80 and Intersil 6100 microprocessor, four PIAs for control and several multiplexers and demultiplexers. After a power on-reset, the executive 6502 has full control of the 460Z and the bus beyond which it can access by mapping a 4K porthole through the 460Z's address space. The 6502 can, of course, load and examine memory in this area. The 6502 has full control of each line of the Z-80 and 6100 and can bring these processors up in either a single step or full speed mode of operation. Even when these processors are running at full speed, the 6502 can monitor system signals. Thus, the 6502 can trap certain instructions, stop the host processor, perform some operations and resume operation of the host. Thus, instructions such as I/O and absolute memory references can be micro-programmed. This allows the user to relocate programs and modify I/Os on a general basis, i.e., without modifying each program.

Other side benefits of this architecture are that the 6502 can disconnect itself from the 460Z's internal bus and go on to other tasks – allowing true multiprocessing. The 6502 can easily disassemble programs running on the 460Z since it can read all signal lines and can single step the processors. The 460Z has provisions for a third processor such as a 2650 or F8, or a new processor that hasn't even been invented yet.

The architecture of the 460Z allows you to expand a 400 system without limit to any number of processors and any amount of memory with full multiprocessing capability.

The OSI 460Z, like most other 400 series boards, is only $29 bare with manual, and, as an introductory special, we are offering this package: OSI 460Z Board Bare with Manual, Intersil IM6100I and Zilog Z-80, all for $99.00. Ohio Scientific Instruments, 11579 Hayden St., Hiram OH 44234.

SWTPC CT-64 Terminal System.

THE BASIC FORUM

from page 11

Digital Group system and is looking forward to getting his hands on their Maxi-BASIC interpreter. For now, his questions concern mainly the technique of handling data with a mass storage device such as a tape or disk.

1. Is it possible to write a BASIC program to ask the terminal operator for data, do some number crunching on that data, then store it on disk or tape?

The answer to this question is certainly yes, but as you would expect, it depends on the BASIC interpreter selected. The ability to directly save data is not generally available on 4K or 8K BASICs, although at least one we know of (BASIC ETC written by the author) permits limited data storage on cassette tape. Some of the 12K extended BASICs allow arrays to be saved directly on cassette

tape, but these still fall short of a true data management capability. For this, it is necessary to have one of the specific tape or disk BASICs. These generally require 16K or more memory and, of course, a computer controllable mass storage device. At this stage in personal computing, cassette/cartridge tape drives and flexible (or so-called floppy) disk drives are the most popular. The pages of *Kilobaud* reflect a number of manufacturers who sell such equipment and software. It is not possible to detail the individual packages here, but most provide the user with the capability of storing the contents of program variables on a tape or disk file for future retrieval.

2. Are there standard BASIC commands for storing variables on mass storage devices?

Although we are unaware of any real standard, there are a few statements that seem to be popular. OPEN is generally used to create a file and set up a temporary memory buffer for the data to be handled. CLOSE is used to release the buffer memory and terminate the file operation. Data is obtained from the mass storage device with the GET statement. The data is placed in variables according to some prearranged scheme. In Altair Disk BASIC, for instance, the variables are associated with character string fields set up previously within the data record on the disk. Other BASICs permit data to be stored without conversion to string fields; i.e., floating point numbers are stored in their internal binary form. The latter technique is also popular with tape systems using BASIC (IBM 5100 for instance). The PUT statement is used to place the contents of variables on the mass storage device. It, of course, complements the GET statement – what you PUT today you can GET tomorrow.

3. Is it possible for one BASIC program to load another from disk or tape?

Most certainly. This capability is available on most tape and disk BASICs intended for the small computer user. The ability to link or chain programs together is important. Large programs can be subdivided and brought into limited memory piece by piece. Data is passed from one segment of the program to the other using mass storage files. While such operations may not constitute true subroutine calls, their usefulness is considerable.

Verlynn has two additional comments which we will reprint for our readers: "In the first issue of *Kilobaud*, many authors kept repeating the idea that programs without documentation were not very understandable. True, but *also* documentation without programs is just as difficult. A professional programmer may understand an abstract description of a programming technique due to his experience with similar situations, but most hobbyists would have a hard time unless specific *non-trivial* examples are included with the documentation. The words just don't have any meaning because we have nothing to relate them to. I find that I can learn more by puzzling out a well-documented program than I can by reading a description of that program.

"Would you or *Kilobaud* (or readers of BASIC Forum) be interested in a Chess program written in Digital Group Z-80 Maxi BASIC. I believe DGZ80MB is pretty close to standard BASICs and it shouldn't be too hard to convert. I will not use the user-defined functions but will use substrings.

"The program is currently under development and should be ready in about a month. I hope to support all the standard rules on chess, including castling, capture en passe, etc. I would be willing to write a complete line by line description of program logic and a table of variables with their uses.

"I think developing a *smart* Chess program would be a good group project for a club or perhaps as a continuing project in *Kilobaud*. I'd willingly provide the seed."

Finally we come to a letter received very recently which contains some sobering food for thought. The contents are particularly pertinent in this day of the ever-expanding, ever-extending BASIC language. Richard Blumenfeld, 3 Marlin Road, Brewster NY 10509, writes: "I have been reading over the latest issues of *Kilobaud*, looking at everybody's suggestions as to how to improve BASIC, or make it more standard. But, everybody seems to be forgetting what BASIC really means: *Beginner's* All-Purpose Symbolic Instruction Code. Notice that I emphasized the word *Beginner's*. The whole purpose behind BASIC was to create a language that would be much easier than FORTRAN or COBOL, or the other high-level languages. All these suggestions are making BASIC much too complicated for the beginner. I think the list of statements on page 23 of the May *Kilobaud* should be included in "standard" BASIC. However, once all these improvements being tossed around are implemented, I think that the new language created should have a new name. Calling it BASIC is almost self-contradictory; it will no longer be a beginner's language."

If you have some BASIC ideas to pass along to the readers of *Kilobaud*, please send them to:

Dick Whipple
PO Box 7082
Tyler TX 75711.

KILOBAUD KLASSROOM

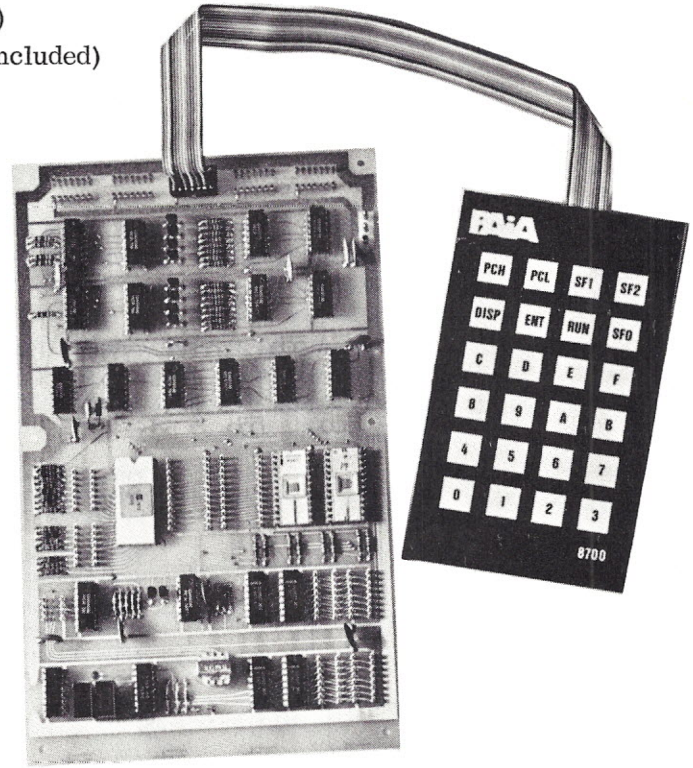
No "Kilobaud Klassroom" This Month

Because of a technical problem "Kilobaud Klassroom" does not appear in this month's issue. It will resume next month. Our apology for this omission.

THE PAIA 8700 COMPUTER/CONTROLLER

An exceptional price on an applications oriented 6503 based micro-processor system featuring:

- 1K bytes RAM locations (512 bytes supplied)
- 1K bytes ROM locations (256 byte monitor included)
- 2 - 8 bit input ports.
- 2 - 8 bit output ports (1 latched, 1 buffered.)
- 24 Key touch operated keypad (used by the monitor to allow entry and execution of user programs - also user definable.)
- 2 - latched seven segment displays (used by monitor to display memory location and contents - easily user programmed.)
- Optional CASSETTE INTERFACE ($22.50) fits entirely on the processor board.



The ideal, low cost solution to implementing all those wild computer based control systems you've been dreaming of!

PAIA software currently available or under development includes:

- Home applications package including: Multi-zone fire and burglar alarm, real time clock, energy saving heat and air conditioning control, computer generated "door-bell".
- Electronic music synthesizer interface.
- Model Railroad controller.
- More..........

8700 COMPUTER/CONTROLLER KIT $149.95

(requires 5v. @ 1.2A.; 12v. @ 150 ma.)

Shipped direct to you by PAIA (add $3.00 postage & handling)

Also available at FULL-LINE Computer stores.

DETAILS ON MORE PAIA KITS IN OUR FREE CATALOG........ WRITE FOR IT TODAY! .. OR PHONE OUR 24HR DEMO/ORDER LINE (405) 843-7396

BAC & MC ORDERS ALSO ACCEPTED.

PAIA ELECTRONICS, INC. Dept. 7-K 1020 W. Wilshire Blvd. Oklahoma City, OK 73116

Cassette I/O Format

... standards are still needed !

A. H. McDonough
M. P. Hammontre
402 Concord St.
El Segundo CA 90245

Without a doubt, one of the areas lacking standardization is cassette recording formats. I'd be very interested in getting feedback from readers who use the format described in the following article. Is it easy to use or difficult? Is it too complicated or too simple? How does it compare to the format(s) you're now using? Do you have any improvements to suggest? Should it be pushed and promoted as a standard format? – John.

This article describes programs which read and write magnetic tape cassette records in a format that has been proposed as a standard for microcomputers. These programs were developed to run on an Altair 8800 with a Tarbell cassette interface. Flowcharts and program listings are included.

The growth of microprocessor/microcomputer technology has been explosive. New devices, new manufacturers, new publications and vast numbers of new users spring up every week. The publications list an endless and bewildering array of new applications and programs, ranging from the most trivial to the most sophisticated, generated by sources ranging from government research institutions to independent seventh graders.

This diversity is an encouraging sign of a flouishing technology but it can also lead to chaos. Some standards are needed to impose order and to provide a medium for the communication of new ideas. As yet no standards committee has any appreciable influence in the microcomputer field and there is no industry giant, no micro-IBM, to impose de facto standards. It remains for the individual user or groups of users to propose standards they believe useful, and for the acceptance or rejection by other users to determine what is truly a standard.

The Proposed Format

One potentially important and useful standard was proposed by Charles H. Eby in a recent issue of *SCCS Interface Magazine*[1]. This was a standard format for use on magnetic tape cassettes. The basic format is shown in Fig. 1 and Fig. 2. This format is tailored to the Tarbell cassette interface and the START byte and SYNC byte are dictated by the requirements of that interface.

File marks are written both before and after the

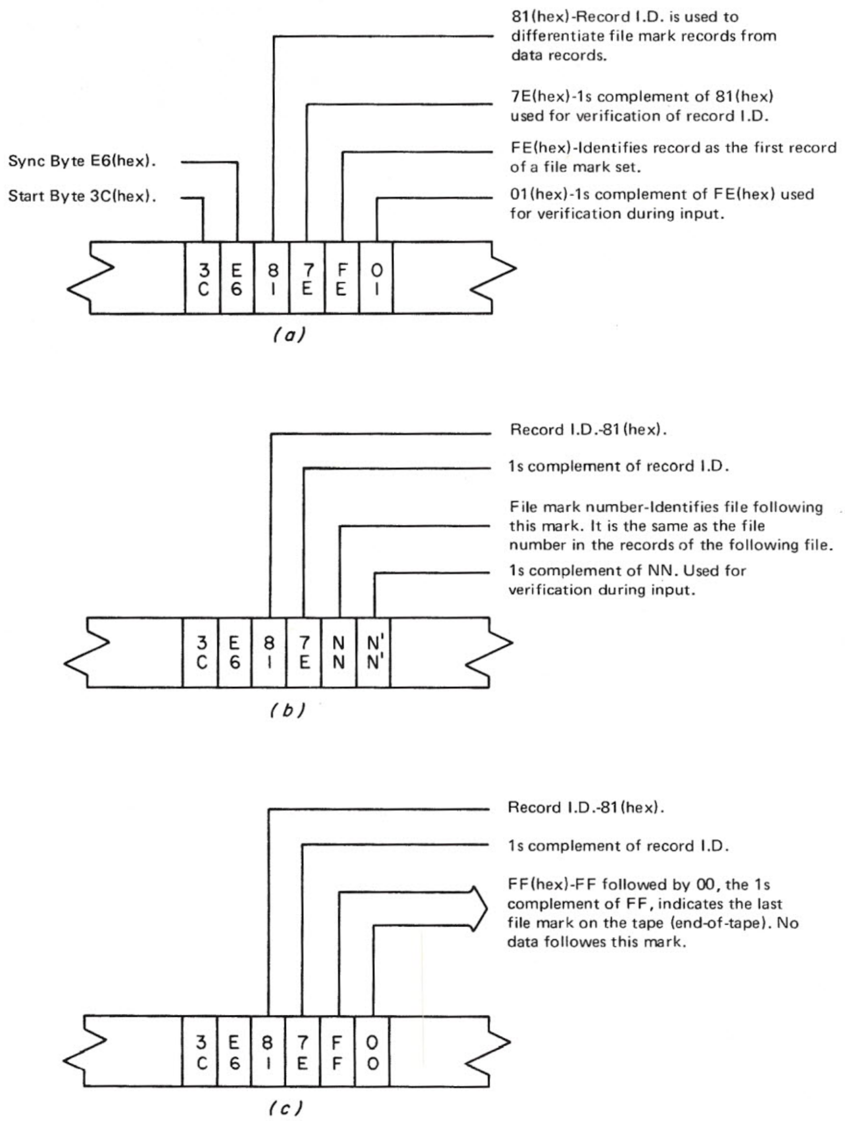


Fig. 1. (a) File Mark Record 1, (b) File Mark Record 2, (c) File Mark Record 2 End-of-tape.

data records of the file. The beginning file mark consists of two physical records as shown in Fig. 2b. A special end-of-tape mark indicates the end of the last file on the tape. This end-of-tape mark differs from an end-of-file mark only in using FF (hex) as a file mark number. Complete details of the recording format can be found in the referenced article.

The programs described in this article are based on the assumption that each tape contains only a single file. This was done for easier record keeping and for operational convenience. This restriction is not inherent in the programs which could easily be modified to allow multiple files on a single cassette.

Four separate programs are required to handle the reading and writing of tapes. These are a Cassette Output Boot Program, a Cassette Bootstrap Program, a Cassette System Input Routine and a Cassette System Output Routine. Flowcharts for these programs are shown in Fig. 3 through Fig. 6 and program listings are given in Table 1 through Table 4. The first of these, the Cassette Output Boot Program, is used only one time in normal operation. It is used to create tape files which contain the Cassette System Input and Output Routines. The Cassette Output Boot Program is loaded into the computer manually or from another medium, such as paper tape. The Cassette System Input and Output Routines are loaded in the same manner. The Output Boot is then activated and used to write the other two programs onto tape. The Output Boot Routine is never used again unless, for some reason, new tapes of the Input and Output Routines must be generated.

Cassette Output Boot Program

The flowchart for the Cassette Output Boot Program is shown in Fig. 3 and the listing is given in Table 1. The loop which checks cassette status and writes to tape is designated COUT in Table 3. The sub-loop which continues to check cassette status until it receives an indication that the cassette is ready is designated CLOP.

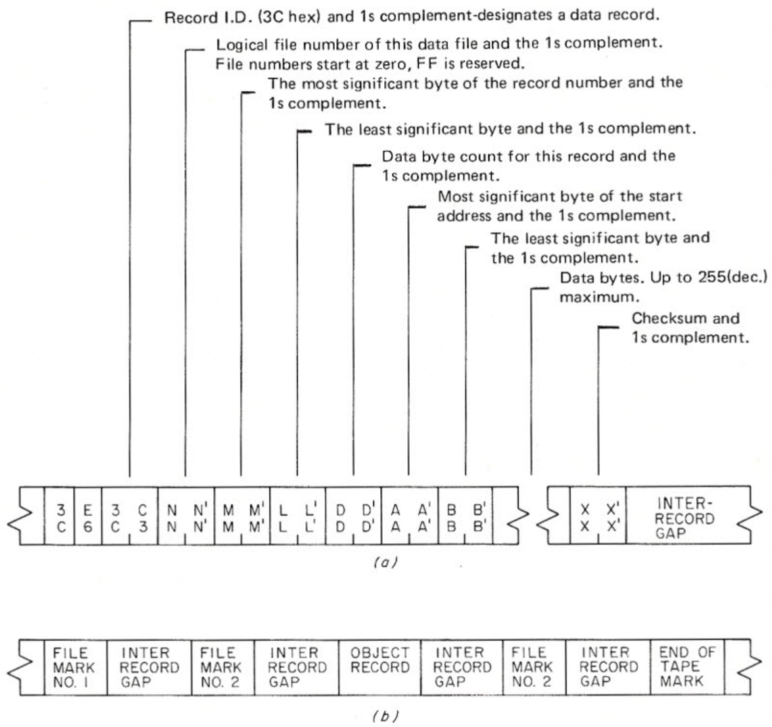


Fig. 2. (a) Object record, (b) typical file with single object record.

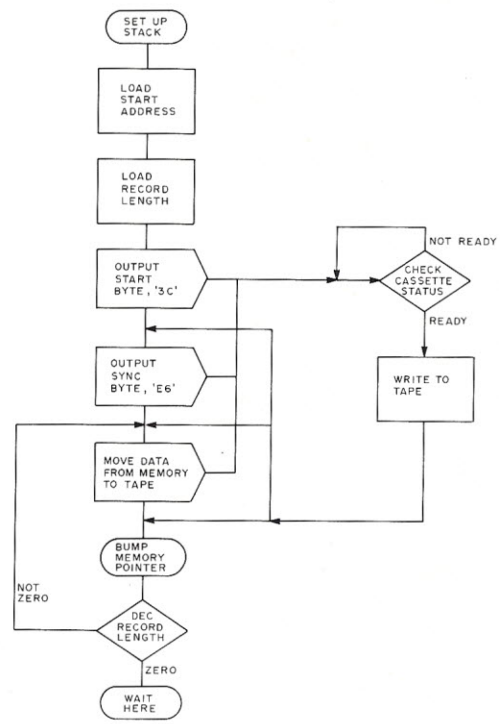


Fig. 3. Cassette output boot program flowchart.

Cassette Bootstrap Program

Once the Cassette System Input and Output Routines are on the tape the Cassette Bootstrap Program may be used to call them into memory. A flowchart for the Cassette Bootstrap Program is shown in Fig. 4 and a listing is given in Table 2. It is assumed that the Cassette System Input and Output Routines are the only things on the tape being loaded. As a result there is no need to load a record length which can be decremented for each byte read until a zero condition signals the end of the record. A more general form is shown in Fig. 4a, where a record length is entered and decremented for each input operation and, when the value reaches zero, the system enters a WAIT state.

The Cassette Input and Output Routines are loaded into the computer from the tape by the Cassette Bootstrap Program. This program may be loaded manually, loaded from another medium, such as paper tape, or may be in a ROM as a permanent part of the system. The program uses a loop to check the status of the Tarbell cassette interface and, when the interface is ready reads the next data byte. This section of the program is labeled LOOP.

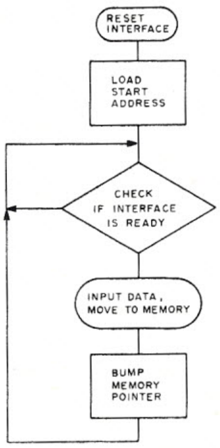


Fig. 4. Cassette bootstrap program flowchart.

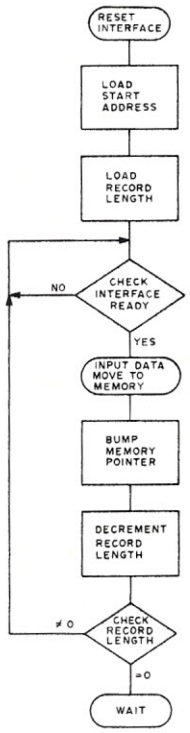


Fig. 4a. Cassette bootstrap program flowchart.

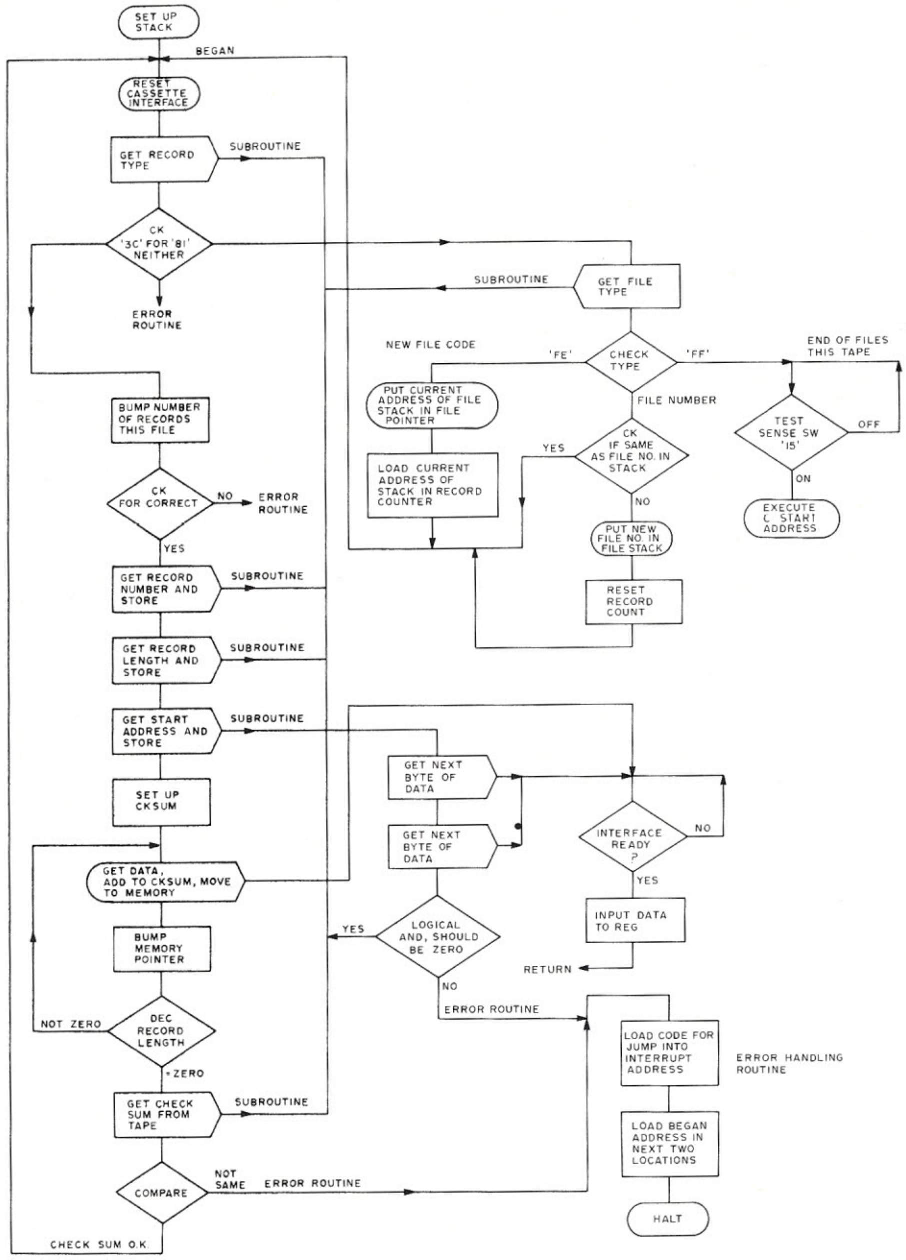


Fig. 5. Cassette input routine flowchart.

Cassette System Input Routine

The Cassette System Input Routine and the Cassette System Output Routine can now be used to read and write tapes in the standard format. Consider first the Input Routine. The flow diagram is shown in Fig. 5 and the code listing is given in Table 3.

There are two basic elements of this routine. The first is a loop that checks cassette interface status and, when the interface is ready, reads in one byte from the tape. The loop is essentially the same as the loop in the bootstrap routines. The second element is a subroutine that executes this loop twice. The loop is labeled INPUT and the subroutine is labeled GET 2. The routine reads the first two characters (first byte) and checks to determine whether these characters are 3C (hex) or 81 (hex). The first of these indicates an object record and the second indicates a file mark record.

If the record is an object record the routine retrieves from memory the address where the previous record number is stored, and then retrieves and increments the previous record number. It then executes the GET 2 subroutine to read in the file number from the tape. This is compared against the file number in memory and, if they are not the same, the routine goes to the ERROR subroutine.

If the file number is correct the routine then reads in and stores the physical record number, the data byte count, and the start address and sets Register C, the checksum register, to zero. At the conclusion of these operations the data byte count is in Register B, and the start address is in Registers H and L. The routine then initiates repetitive input operations using the INPUT subroutine. After each input operation the memory address in H and L is incremented and the byte count in Register B is decremented. At each input operation the data is added to the checksum in Register C.

After all data has been read (byte count decremented to zero) the checksum is read from tape and

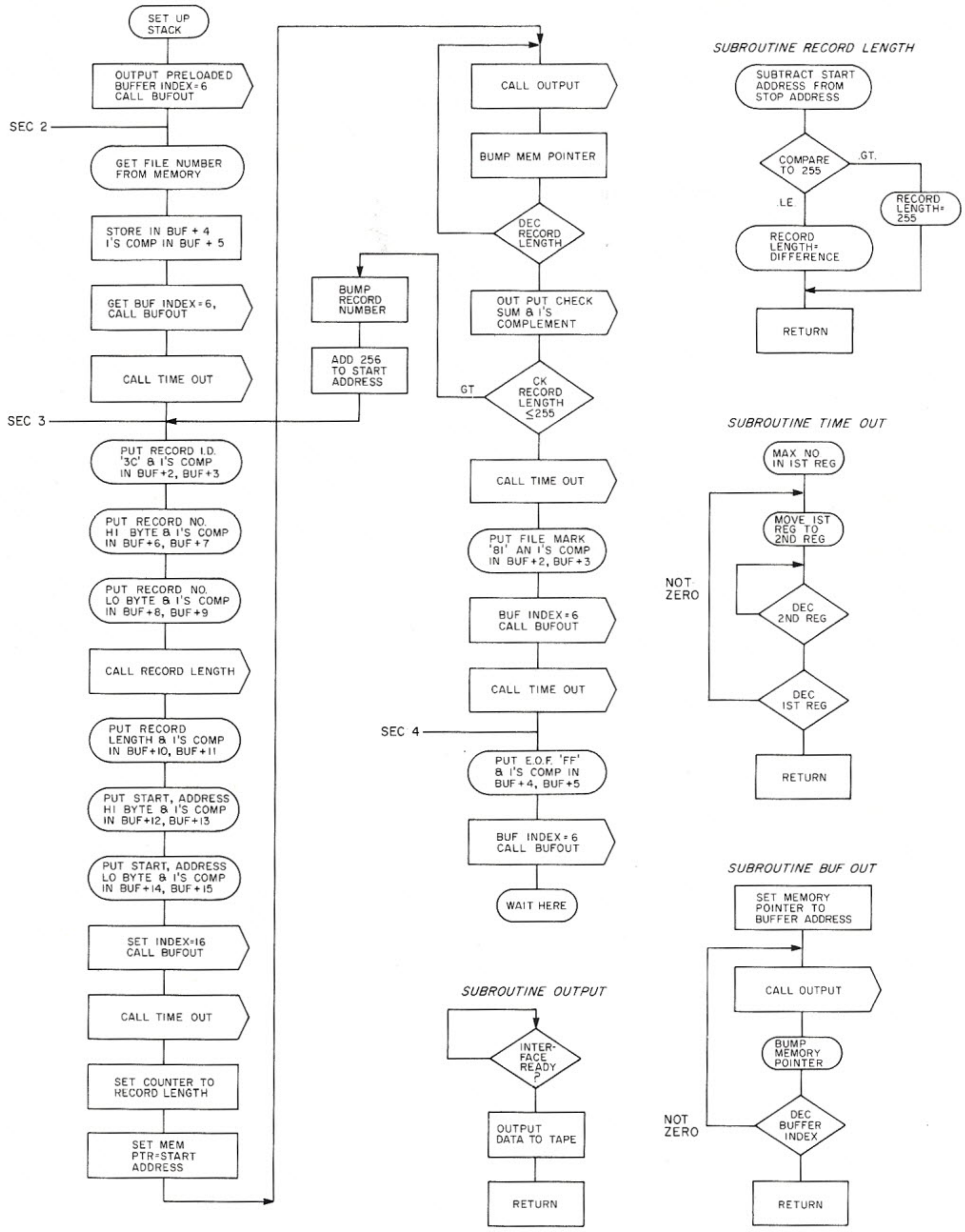


Fig. 6. Cassette system output routine flowchart.

compared to the checksum in Register C. If the two are not equal the routine exits to the ERROR subroutine. If the two are equal the routine goes back to BEGAN and reads the next record. The ERROR subroutine simply puts the JUMP instruction in octal address 070 and the address of BEGAN in locations 071, 072. The system then halts. This allows the operator to rewind the tape and restart the input operation via a manual interrupt.

If the record tape is a file mark record, indicated by 81 (hex) rather than 3C (hex) as the first two characters of the record, the Input Routine jumps to a file handler subroutine labeled FILEH. The file handler first uses the GET 2 subroutine to read the next two bytes from tape. The routine compares these bytes with FE (hex), the code which indicates the start of a file mark set. If the two are equal the routine dumps to the location labeled HEAD. This subroutine loads the Registers H and L with the address of the file stack. The address of the file stack is then stored in memory at the location designated as containing the file pointer (FLPTR). The file pointer is now set to the address of the file stack. This operation is then repeated for the record stack. At the conclusion of this operation the location labeled RECNT contains the address of the record stack. These are housekeeping operations required for starting a new file. The subroutine then returns to BEGAN and reinitiates execution.

The second record normally handled would be the second record of a file mark set, shown in Fig. 1b. The detection of 81 (hex) again hands execution over to FILEH. This time, however, there is no FE (hex) to designate the start of a file mark set. FILEH will then test for the end-of-tape mark indicated by FF (hex) file mark number. If this is not present there is only one possible case remaining. The file mark record is the second record of a file mark set. This may occur either at the beginning of a file or as the echo mark at the end of a file.

The file handler routine loads the file pointer in Registers H and L. It moves the file number which was read from tape into Register B from Register A. It then loads into Register A the contents of the memory location, the address of which is in Registers H and L. As a result the file number from the file stack is in Register A and the file number from tape is in Register B. The routine then compares the two. If the two are equal this indicates that the file mark record is the echo mark at the end of an object record since the file number has already been entered by the second file mark record at the start of the file.

This is clear if the case where the two are not equal is considered. In this case the file pointer is incremented and the file number read from the tape (stored in Register B) is moved to memory to the location indicated by the incremented file pointer. The correct file number is now in the file stack. When a new file number is first read into the computer the file number read from tape cannot agree with the file number in the stack since it is the reading in of the file mark record that sets the proper file number. The two can only agree when an echo file mark is read following an object record. In this case the correct file number has already been established by the file mark

record at the beginning of the file.

After the file pointer and the file number have been set correctly the file handler updates the record pointer. In all cases except the end-of-tape mark the routine returns to BEGAN to find and read the next record.

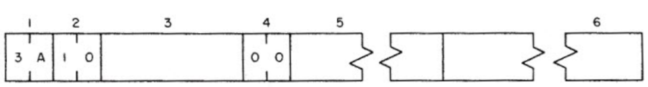
Cassette System Output Routine

The Cassette System Output Routine is used to write object records and the appropriate file mark records on the cassette tape. This routine is shown in flowchart form in Fig. 6 and a listing is given in Table 4. The user must manually load the start and stop addresses defining the block of data to be recorded and must also manually load the file number.

This routine is long but relatively straightforward. A buffer in memory, beginning at location BUF, is preloaded with all the basic characters required to generate the first file mark record. The first section of the Output Routine uses Register B as an index, loads the register with the number six, and then transfers control to the BUFOUT subroutine. This subroutine outputs to the tape the first N characters from the buffer, where N is the number in Register B. These six characters, all in hex, are: 3C (tape start), E6 (tape sync), 81, 7E, FE, and 01. The first two bytes, 3C and E6, are control bytes used by the cassette interface and are not recorded. The characters that are recorded constitute the first file mark record of a file mark set, shown in Fig. 1a. Control then returns to the main routine which calls the subroutines labeled TMOT. This subroutine is a simple timing device to generate the necessary inter-record gap.

The second section of the Output Routine generates the second file mark record shown in Fig. 1b. The file number, which has been loaded manually, is brought into Register A and then loaded into the buffer at BUF+5. The BUFOUT and TMOT subroutines are then called and they write the second file mark record on tape and generate the second inter-record gap.

The third section of the Output Routine generates the object records. It begins by restructuring the contents of the buffer so that the buffer contains the necessary header characters which precede the data bytes. A 3C (hex) is loaded at BUF+2 and the 1s complement, C3 (hex) is loaded at BUF+3. These denote that the following record is an object record. The file number is already in the buffer so the routine next handles the physical record number. The most significant byte of the record number (MRPN) is loaded in Register A and then moved to BUF+6. The 1s complement is generated and loaded at BUF+7. The least significant byte of the record number and its 1s complement are loaded into BUF+8 and BUF+9 in similar fashion. The Output Routine then calls a record length subroutine designated RECLGN.



Block 1; is ASCII "colon" and is used for a record identifier.

Block 2; contains hex value up to 10, this designates record length of this record.

Block 3; contains start address of this record.

Block 4; not used but must be accounted for.

Block 5; data contained in this record.

Block 6; contains 1s complement of ALL parts of record except record identifier.

Fig. 7. INTEL tape format.

The RECLGN subroutine subtracts the start address from the stop address to determine the number of data bytes to be recorded. This number is compared against 255 (dec.), the maximum number of bytes in a single object record. If the size of the block is greater than 255 bytes then the block must be broken up into two or more object records, and the data byte count is set to 255. If the block size is less than 255 then the data byte count (DBC) is set to the actual block size. Control is then returned to the main routine. The data byte count is stored in BUF+10 and the 1s complement is stored in BUF+11. The Output Routine then fills the remainder of the buffer with the most significant byte of the start address, its 1s complement, the least significant byte of the start address and its complement, completing the header for an object record. The routine then calls BUFOUT which writes the header on the cassette tape.

The routine then outputs the data bytes. The data byte count is set into Register B, the start address into Register A and Register C, which will be used to hold the checksum, is set to zero. As each byte is written out to the tape the address (Reg. A) is incremented, the data byte count (Reg. B) is decremented, and the byte is added to the checksum (Reg. C). When the data byte count reaches zero the Output Routine writes the checksum on tape and determines whether to write another object record or an end-of-file record. Since there is only one file per tape the end-of-file record is immediately followed by an end-of-tape marker.

The data byte count is brought into Register A and compared to 255. If DBC $\leqslant$ 255 then the object record is the last record (or perhaps the only record) of this file. If the DBC $>$ 255 then at least one more object record must be generated. If another object record must be produced the start address is increased by 255, the physical record number is incremented by one, and control is returned to Section 3 of the Output Routine to generate the next object record.

If the DBC $\leqslant$ 255 then the Output Routine rearranges the buffer, entering 81 (hex) at BUF+2 and the 1s complement at BUF+3, recreating the second file mark record, and writes the record on the tape following the object record. The routine then prepares an end-of-tape marker. This is done by entering FF (hex) at BUF+5 and the 1s complement at BUF+6. The BUFOUT subroutine then writes this marker on tape. This completes the recording and the system enters the WAIT state.

As previously stated this system assumes that each cassette is to contain a single file. However, any user who wishes to record multiple files on a single tape can easily do so. As a simple illustration the operator can use the tape counter to keep track of the location of several files on the tape and let the system believe that each new segment of tape is, in fact, a new tape. If this procedure seems troublesome then relatively minor modifications will allow this system to handle multiple files.

Comparions with Other Formats

Several tape formats were investigated before the decision was made to implement this specific format. The first and most obvious option to be considered was

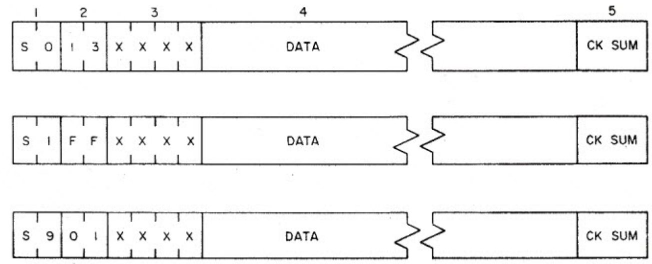
the INTEL format; originally a paper tape format but adopted by many users for magnetic tape cassettes. The basic outline of the INTEL format is shown in Fig. 7.

One severe limitation of this format is the maximum record length of 16 bytes (hex 10). A start address must be setup every 16 bytes, the record identifier must be checked every 16 bytes, a byte count must be re-established every 16 bytes the checksum must be handled every 16 bytes. The 16-byte limit is so small that very few programs or data sets can be handled in a single record. The end result is high overhead and considerable inefficiency. The number 256, the basic record length limit in the proposed format, represents one page of 8080 memory, the capacity of one ROM, and the maximum count of a full register.

A more severe disadvantage of the INTEL format for magnetic tape use is the lack of identifiers. Since there is no file number or record number associated with the data block the system cannot be expanded to provide search capability. The proposed format, on the other hand, includes both file and record identifiers. As a result the system can easily be expanded to include both tape search or memory search.

While the INTEL format is not particularly well-suited to magnetic tape, it offers some considerable advantages when applied to punched cards. Each record will fit on a single punched card, and each card contains its own start address. The cards can therefore be read in any order. However, this feature is of little value to the computer hobbyist or to most professionals since it requires the availability of a card punch and a card reader. This implies a sophisticated and costly system and requires storage of card decks rather than tape cassettes.

A second alternative considered was the Motorola format used with the 6800. This format is shown in Fig. 8. This format avoids the record length limitation of the INTEL and allows data blocks of up to 256 bytes, reducing CPU overhead time. The first byte designates the type of record and this allows considerably more flexibility and possibility of expansion than the INTEL format. However, it is more restricted in file management capability than is the proposed format and offers no significant advantage in magnetic tape applications.



Block 1; SO represents file header of start of record.
S1 represents data record indicator.
S9 represents end of file or record mark.

Block 2; record length, could be 256 bytes but is typically 13 to 16.

Block 3; start address of this record.

Block 4; Data.

Block 5; CK SUM which is the 1s complement of the exclusive or of the sum of ALL bytes except the record identifier.

Fig. 8. Motorola 6800 format.

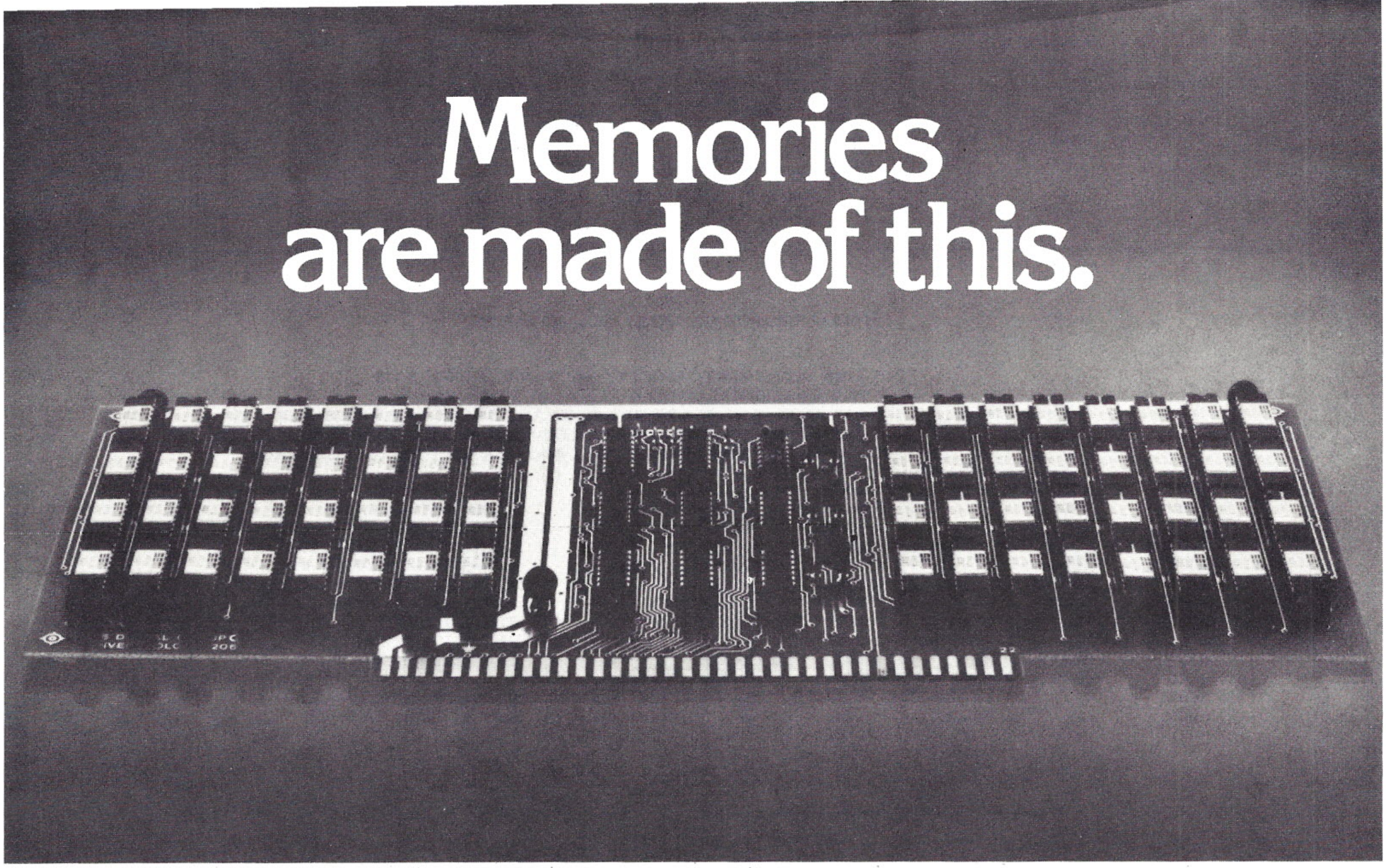
In short, neither the INTEL or Motorola formats were established with magnetic tape in mind. Both are oriented to the single data block on a single piece of punched paper tape (although this is less true of the Motorola 6800 format). Both were established before the use of magnetic tape for microprocessor storage was seen to be economically attractive. As a result neither is readily expandable to include the file search and management techniques which are a significant part of the advantages of magnetic tape storage. ■

Reference

Eby, Charles H., "Cassette Tape Format Standards," *SCCS Interface*, vol. 1, issue 7, June 1976, 44-46.

Table 1. Cassette output boot program listing. *(Continued on page 26.)*

```
                          ;          THIS IS THE 8080 BOOT OUTPUT ROUTINE
                          ;          IT WILL OUTPUT A START CODE AND A SYNC CODE AND THEN
                          ;          OUTPUT ALL DATA LOCATED IN MEMORY FROM THE START ADDRESS
                          ;          TO THE END OF THE RECORD LENGTH.
                          ;          IT IS THE RESPONSIBILITY OF THE USERS OF THIS PROGRAM
                          ;          TO LOAD BOTH THE START ADDRESS AND THE RECORD LENGTH IN THE
                          ;          CORRECT ADDRESS LOCATIONS, USING THE 8080 ADDRESSING SYSTEM OF
                          ;          LOW BYTE FIRST FOLLOWED BY THE HI BYTE
                          ;
0005                      STAK:      DS      5
0005      310500                     LXI     SP,STAK+5          ;SET STACK POINTER
0008      2A3700                     LHLD    SA                 ;LOAD DIRECT, THE START ADDRESS
000B      01F401                     LXI     B,500              ;THIS VALUE IS THE RECORD LENGTH
                          ;          IT MUST BE CHANGED TO FIT EACH RECORD YOU USE
000E      3E3C                       MVI     A,3CH              ;START BYTE CODE
0010      CD2B00                     CALL    COUT               ;OUTPUT START BYTE TO CASSETTE
0013      3EE6                       MVI     A,0E6H             ;SYNC BYTE CODE
0015      CD2B00                     CALL    COUT               ;OUTP TO CASSETTE
0018      7E              LOOP:      MOV     A,M                ;GET DATA FROM MEMORY
0019      CD2B00                     CALL    COUT               ;OUTPUT TO TAPE
001C      23                         INX     H                  ;BUMP POINTER
001D      0B                         DCX     B                  ;DECREMENT RECORD LENGTH
001E      3E00                       MVI     A,0                ;A REG = 0
0020      B8                         CMP     B                  ;CHECK TO SEE IF ALL RECORDS COPIED
0021      C21800                     JNZ     LOOP               ;IF NOT, REPEAT
0024      B9                         CMP     C
0025      C21800                     JNZ     LOOP
0028      C32800          WAIT:      JMP     WAIT               ;WILL LOOP HERE WHEN FINISHED
002B      F5              COUT:      PUSH    PSW                ;PUT A REG IN STACK
002C      DB6E            CLOP:      IN      CASC               ;CHECK CASSETTE STATUS
002E      E620                       ANI     20H                ;CLEAR ALL BUT BIT 5
0030      C22C00                     JNZ     CLOP               ;IF NOT ZERO, CHECK AGAIN
```



32K. One Card. One low price. Only from the Digital Group.

Now, on only one *fully static* card, the Digital Group has squeezed in a whopping 32K of memory. Which, with a little quick addition, means a full 64K architecture now requires only 2 boards instead of 8. That's a 4-to-1 space reduction . . . and leaves one extra memory slot on the Digital Group's standard motherboard still available for future products.

All this and one low price, too.
It just may be the best news of all. Our full static, assembled and tested 32K memory board is only $995. Now that's worth remembering. It's substantially less than our equivalent assembled 8K board prices. (Please note: We're initially offering this 32K board assembled only, but kit versions will soon be available, too—at even lower prices.)

Here's what you get.

Specifications:
- 32K on single card
- Speed—450ns. All of our current CPUs will operate at full-rated speed.
- Decoding—Lower or upper 32K bank
- Power— +5V only @ 4A
- Card size—12″ x 5″ (excluding connector fingers)

Features:
- May be intermixed on Digital Group systems with our 8K memory cards
- All data and address lines are buffered
- *Fully* static memories—EMM 4801 (450ns) or equivalent

Price:
32K board complete, assembled and tested $995.00

For all the memorable details, just fill out the coupon below. (Then all you have to remember is to mail it in.)

the digital group

P.O. Box 6528
Denver, Colorado 80206 (303) 777-7133 **D12**

☐ **I promise to mail this in, so add me to your mailing list!**

Name ______________________________

Address ______________________________

City/State/Zip ______________________________

☐ Remember me? I'm already on your mailing list, but I need the memory spec sheet desperately.

```
0033    F1                POP    PSW         ;RETRIEVE A REG FROM STACK
0034    D36F              OUT    CASD        ;OUTPUT A REG TO TAPE
0036    C9                RET                ;RETURN
0037    C002      SA:     DW     2C0H        ;THE START ADD TO BE LOADED HERE
006F              CASD    EQU    6FH
006E              CASC    EQU    6EH
```

Table 2. Cassette bootstrap program listing.

```
                  ;       THIS IS THE 8080 CASSETTE BOOT STRAP PROGRAM
                  ;       IT WAS ASSEMBLED TO RUN AT LOCATION ZERO
                  ;       AFTER IT IS IN, START THE TAPE AND START THE RECORDER
                  ;       THE TAPE PROGRAM WILL BE LOCATED AT THE ADDRESS THAT WAS
                  ;       MANUALLY INSERTED BY YOU AT LOCATION 'SA' (SEE LISTING)
                  ;
0000    3E10              MVI    A,10H       ;SET BIT 4 OF A = 1
0002    D36E              OUT    CASC        ;RESETS INTERFACE
0004    2A1500            LHLD   SA          ;PUTS STARTING ADD IN H & L
0007    DB6E      LOOP:   IN     CASC        ;READS INTERFACE STATUS
0009    E610              ANI    10H         ;CLEARS ALL BUT BIT 4
000B    C20700            JNZ    LOOP        ;WAIT IN LOOP UNTIL READY
000E    DB6F              IN     CASD        ;READ A DATA BYTE
0010    77                MOV    M,A         ;STORE IN MEM
0011    23                INX    H           ;BUMP MEM POINTER
0012    C30700            JMP    LOOP        ;GET NEXT DATA
0015    C002      SA:     DW     2C0H
006E              CASC    EQU    6EH         ;CASSETTE STATUS PORT
006F              CASD    EQU    6FH         ;CASSETTE DATA PORT
                          END
```

Table 3. Cassette system input routine listing.

```
                  ;       THIS IS THE TAPE INPUT ROUTINE FOR THE INTEL 8080
                  ;       MICROPROCESSOR USED IN CONJUNCTION WITH A TARBELL CASSETTE
                  ;       INTERFACE AND AN AUDIO CASSETTE AS REF IN THE JUNE
                  ;       1976 ISSUE OF INTERFACE MAGAZINE (PG 43).
                  ;       IF A CHECK SUM ERROR ACCURS THE PROGRAM WILL HALT. MOVE THE
                  ;       TAPE BACK AND RESTART IT. THEN HIT THE INTERRUPT SWITCH.
                  ;       EACH RECORD HAS ITS OWN START ADDRESS SO YOU MAY START
                  ;       ANY PLACE. THIS MAY CAUSE AN ERROR IN THE RECORD COUNT HOWEVER.
                  ;       IF THE TAPE HAS BEEN READ WITHOUT ERRORS IT WILL PUT ITS
                  ;       SELF INTO A TIGHT LOOP. IF YOU WANT TO EXECUTE AT THE
                  ;       START ADDRESS, PUSH SENSE SWITCH 15 UP.
                  ;
FF0F    31FFF8            LXI    SP,STACK    ;PUT THE POINTER AT THE TOP
FF12    3E10      BEGAN:  MVI    A,10H       ;SET BIT 4 OF A=1
FF14    D36E              OUT    CASC        ;RESET INTERFACE
FF16    FB                EI                 ;ENABLE INTERRUPTS
FF17    CD99FF            CALL   GET2        ;PULLS IN TYPE I.D.
FF1A    FE81              CPI    81H         ;CHECKS FOR FILE I.D.
FF1C    CA71FF            JZ     FILEH       ;FILE HANDLER
FF1F    FE3C              CPI    3CH         ;RECORD HANDLER
FF21    C2B0FF            JNZ    ERROR       ;ERROR HANDLER
FF24    2AD2FF            LHLD   RECNT       ;NUMBER OR RECORDS THIS FILE
FF27    34                INR    M           ;BUMP BY ONE
FF28    CD99FF            CALL   GET2        ;GET FILE NUMBER
FF2B    47                MOV    B,A         ;TEMP STORAGE
FF2C    2AD4FF            LHLD   FLPTR       ;ADDRESS IN FLOPTR, NOW IN H,L
FF2F    BE                CMP    M           ;CHECK TO SEE IF SAME
FF30    C2B0FF            JNZ    ERROR       ;IF NOT, JMP TO ERROR
FF33    CD99FF            CALL   GET2        ;HI BYTE OF REC#
FF36    32CEFF            STA    HIBYTE      ;STORES IT IN 'HI BYTE'
FF39    CD99FF            CALL   GET2        ;LO BYTE OF REC#
FF3C    32CDFF            STA    LOBYTE      ;STORES IT IN 'LO BYTE'
FF3F    CD99FF            CALL   GET2        ;GETS DATA COUNT
FF42    32D1FF            STA    DBCT        ;STORES IT IN 'DBC'
FF45    CD99FF            CALL   GET2        ;HI BYTE OF START ADD
FF48    32D0FF            STA    HSA         ;STORES IT IN 'HSA'
FF4B    CD99FF            CALL   GET2        ;GET LOW BYTE OF START ADD
FF4E    32CFFF            STA    SA          ;STORES IT IN 'LSA'
FF51    3AD1FF            LDA    DBCT        ;DATA BYTE COUNT IN A
FF54    C601              ADI    1           ;ALLOWS FIRST AND LAST LOCATIONS
FF56    47                MOV    B,A         ;NOW IN B
FF57    2ACFFF            LHLD   SA          ;PUT START ADD IN H,L
FF5A    0E00              MVI    C,0         ;RESET CHECK SUM
FF5C    CDA6FF    DATIN:  CALL   INPUT       ;GET DATA
FF5F    77                MOV    M,A         ;STORE IN MEMORY
FF60    81                ADD    C           ;ADD TO CHECK SUM
```

```
FF61    4F                MOV    C,A          ;RETURN TO C RFG
FF62    23                INX    H            ;BUMP MEM POINTER
FF63    05                DCR    B            ;DECREMENT BYTE COUNT
FF64    C25CFF            JNZ    DATIN        ;GET MORE DATA
FF67    CD99FF            CALL   GET2         ;GETS CHECK SUM
FF6A    B9                CMP    C            ;COMPARE CHECKSUM
FF6B    C2B0FF            JNZ    ERROR        ;CHECK SUM ERROR
FF6E    C312FF            JMP    BEGAN        ;START AGAIN
FF71    CD99FF   FILEH:   CALL   GET2         ;GETS TYPE OF FILE CODE
FF74    FEFE              CPI    0FEH         ;FILE HEADER CODE
FF76    CABEFF            JZ     HEAD         ;SET POINTER
FF79    FEFF              CPI    0FFH         ;FILE STOP CODE
FF7B    CAEAFF            JZ     WAIT         ;END OF FILES ON TAPE
FF7E    2AD4FF            LHLD   FLPTR        ;LOAD FILE TABLE POINTER
FF81    47                MOV    B,A          ;STORE FILE #IN B
FF82    7E                MOV    A,M          ;GET PRESENT FILE #
FF83    B8                CMP    B            ;IF SAME, END OF FILE
FF84    CA12FF            JZ     BEGAN        ;GET NEXT RECORD
FF87    23                INX    H            ;BUMP FILE #
FF88    22D4FF            SHLD   FLPTR        ;STORE ADD IN POINTER
FF8B    70                MOV    M,B          ;PUT NEW FILE #IN TABLE
FF8C    2AD2FF            LHLD   RECNT        ;ADD OF RECORD POINTER
FF8F    23                INX    H            ;BUMP ADDRESS
FF90    22D2FF            SHLD   RECNT        ;STORE IN RECORD POINTER
FF93    C312FF            JMP    BEGAN        ;GET NEXT RECORD
FF96    00                NOP                 ;THIS IS THE END OF THE MAIN PROGRAM
FF97    00                NOP                 ;THE FOLLOWING ARE THE VARIOUS
FF98    00                NOP                 ;SUBROUTINES USED WITH IT
                 ;
FF99    CDA6FF   GET2:    CALL   INPUT        ;GET DATA FROM TAPE
FF9C    47                MOV    B,A          ;TEMP STORAGE
FF9D    CDA6FF            CALL   INPUT        ;GETS 1'S COMP
FFA0    A0                ANA    B            ;SHOULD BE ZERO
FFA1    C2B0FF            JNZ    ERROR
FFA4    78                MOV    A,B          ;PUT DATA BACK IN A
FFA5    C9                RET
                 ;
FFA6    DB6E     INPUT:   IN     CASC         ;READS INTERFACE STATUS
FFA8    E610              ANI    10H          ;CLEARS ALL BUT BIT 4
FFAA    C2A6FF            JNZ    INPUT        ;TRY AGAIN
FFAD    DB6F              IN     CASD         ;GET DATA BYTE
FFAF    C9                RET
                 ;
FFB0    3EC3     ERROR:   MVI    A,0C3H       ;JUMP CODE
FFB2    213800            LXI    H,0700       ;INTERRUPT ADDRESS
FFB5    77                MOV    M,A          ;STORE INSTRUCTION THERE
FFB6    1112FF            LXI    D,BEGAN      ;LOAD ADD OF BEGAN IN D,E
FFB9    23                INX    H            ;BUMP MEM POINTER TO 0710
FFBA    73                MOV    M,E          ;STORE LOW BYTE THERE
FFBB    23                INX    H            ;BUMP MEM POINTER TO 0720
FFBC    72                MOV    M,D          ;STORE HI BYTE THERE
FFBD    76                HLT    ;BACK UP TAPE, HIT INTERRUPT
FFBE    21DFFF   HEAD:    LXI    H,FILE-1     ;ADDRESS OF FILE STACK
FFC1    22D4FF            SHLD   FLPTR        ;ADDRESS OF FILE POINTER
FFC4    21D5FF            LXI    H,RECORD-1   ;ADDRESS OF RECORD STACK
FFC7    22D2FF            SHLD   RECNT        ;ADDRESS OF RECORD STACK
FFCA    C312FF            JMP    BEGAN        ;GET NEXT DATA WORD
                 ;
0001             LOBYTE:  DS     1            ;RECORD NUMBER, LOW BYTE
0001             HIBYTE:  DS     1            ;RECORD NUMBER, HI BYTE
0001             SA:      DS     1            ;LOW BYTE OF START ADD
0001             HSA:     DS     1            ;HI BYTE OF START ADD
0001             DBCT:    DS     1            ;DATA BYTE COUNT
FFD2    D5FF     RECNT:   DW     RECORD-1     ;ADDRESS OF RECORD STACK
FFD4    DFFF     FLPTR:   DW     FILE-1       ;ADDRESS OF FILE STACK
000A             RECORD:  DS     10           ;RECORD LENGTH STACK
000A             FILE:    DS     10           ;FILE NUMBER STACK
FFEA    DBFF     WAIT:    IN     SENS         ;READ FRONT PANEL SENSE SW
FFEC    E680              ANI    80H          ;CLEAR ALL BUT BIT 15
FFEE    CAEAFF            JZ     WAIT         ;CHECK AGAIN
FFF1    2ACFFF            LHLD   SA           ;LOAD START ADDRESS INTO H,L
FFF4    E9                PCHL                ;EXECUTE START ADDRESS
00FF             SENS     EQU    0FFH         ;SENSE SWITCHES, FRONT PANEL
006F             CASD     EQU    6FH          ;CASSETTE DATA PORT
006E             CASC     EQU    6EH          ;CASSETTE STATUS PORT
                          END
```

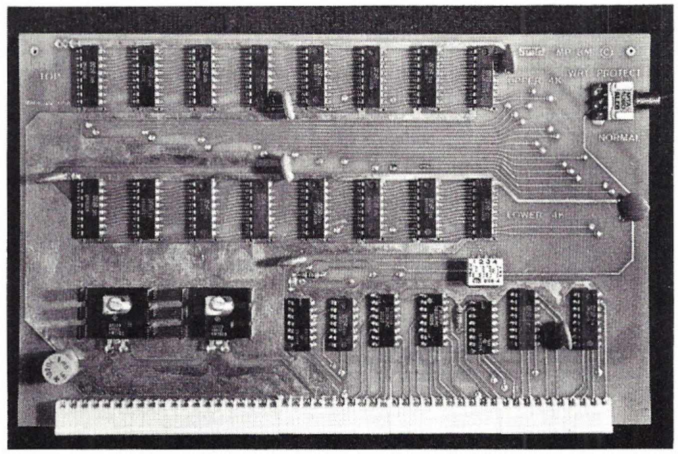
Table 4. Cassette system output routine listing.

```
;       THIS IS THE TAPE OUTPUT ROUTINE FOR THE INTEL 8080
;       MICROPROCESSOR USED IN CONJUNCTION WITH A TARBELL CASSETTE
;       INTERFACE AND AN AUDIO TAPE CASSETTE
;       THIS PROGRAM REQUIRES THE USER TO MANUALLY LOAD THE TWO
;       BYTE START ADDRESS INTO THE LOCATIONS CALLED, MSA & LSA,
;       AND THE TWO BYTE STOP ADDRESS INTO THE LOCATION CALLED MST
```

```
                        ;       & LST. THEIR ADDRESSES ARE F8 00 & F8 02
                        ;       THIS PROGRAM HAS BEEN LOCATED AT THE TOP OF 65 K OF CORE
                        ;       BUT MAY BE REASSEMBLED ANYWHERE. THE ONLY ABSOLUTE ADDRESSES
                        ;       ARE THOSE OF THE INTERRUPTS.
                        ;
F800                            ORG     0F800H          ;FIRST LOCATION OF LAST 2 K OF CORE
0001                    LSA:    DS      1
0001                    MSA:    DS      1
0001                    LST:    DS      1
0001                    MST:    DS      1
F8FF                            ORG     0F8FFH          ;STACK LOCATION
F8FF                    STACK:
FDED                            ORG     0FDEDH          ;THIS WILL PUT THIS PROGRAM AT THE TOP
FDED    31FFF8                  LXI     SP,STACK        ;LD ORIGION OF S.P.
FDF0    0606                    MVI     B,6             ;LD INDEX
FDF2    CDC3FE                  CALL BUFOUT             ;OUTPUT BUFFER TO TAPE
FDF5    CDB7FE                  CALL    TMOT            ;PRODUCES INTER RECORD GAP
                        ;
FDF8    00              SEC2:   NOP
FDF9    2103FF                  LXI     H,BUF+4
FDFC    11FBFE                  LXI     D,FILEN         ;FILE NUMBER ADDRESS
FDFF    1A                      LDAX    D               ;FILE NUMBER IN A
FE00    77                      MOV     M,A             ;IN BUFFER
FE01    23                      INX     H               ;BUMP MEM POINTER
FE02    2F                      CMA     ;1'S COMP
FE03    77                      MOV     M,A             ;STORE IN BUFFER
FE04    0606                    MVI     B,6             ;SET INDEX TO SIX
FE06    CDC3FE                  CALL    BUFOUT          ;OUTPUT BUFFER TO TAPE
FE09    CDB7FE                  CALL    TMOT            ;PRODUCES INTER RECORD GAP
                        ;
FE0C    00              SEC3:   NOP
FE0D    2101FF                  LXI     H,BUF+2         ;ADD OF RECORD I.D.
FE10    3E3C                    MVI     A,3CH           ;CODE FOR RECORD I.D.
FE12    77                      MOV     M,A             ;STORE SAME
FE13    2F                      CMA                     ;1'S COMP
FE14    23                      INX     H               ;BUMP POINTER
FE15    77                      MOV     M,A             ;STORE IN BUF
FE16    2105FF                  LXI     H,BUF+6         ;ADD OF RECORD NUMBER IN BUF
FE19    3AFDFE                  LDA     MPRN            ;HI BYTE IN A
FE1C    77                      MOV     M,A             ;STORE IN BUF
FE1D    2F                      CMA
FE1E    23                      INX     H               ;BUMP POINTER
FE1F    77                      MOV     M,A             ;STORE IN BUF
FE20    23                      INX     H               ;BUMP POINTER
FE21    3AFCFE                  LDA     LPRN            ;IN A REG
FE24    77                      MOV     M,A             ;STORE IN MEM
FE25    2F                      CMA
FE26    23                      INX     H
FE27    77                      MOV     M,A             ;STORE IN BUF
FE28    CDDCFE                  CALL    RECLGN          ;GET RECORD LENGTH
FE2B    3209FF                  STA     BUF+10          ;DATA BYTE COUNT
FE2E    2F                      CMA
FE2F    320AFF                  STA     BUF+11          ;1'S COMP
FE32    3A01F8                  LDA     MSA             ;HI BYTE OF START ADDRESS
FE35    320BFF                  STA     BUF+12
FE38    2F                      CMA
FE39    320CFF                  STA     BUF+13
FE3C    3A00F8                  LDA     LSA             ;LOW BYTE OF SA
FE3F    320DFF                  STA     BUF+14
FE42    2F                      CMA
FE43    320EFF                  STA     BUF+15
FE46    0610                    MVI     B,16            ;16 BYTE LEADER IN BUF
FE48    CDC3FE                  CALL    BUFOUT
                        ;
                        ;       THIS SECTION OUTPUTS DATA, B REG IS COUNT & C REG IS CKSM
                        ;
FE4B    3AFEFE                  LDA     DBC             ;DATA BYTE COUNT
FE4E    C601                    ADI     1               ;ALLOWS FIRST AND LAST LOCATIONS
FE50    47                      MOV     B,A             ;SET UP RECORD LENGTH COUNT
FE51    0E00                    MVI     C,0
FE53    2A00F8                  LHLD    LSA             ;H & L CONTAIN START ADDRESS
FE56    EB                      XCHG                    ;START ADD NOW IN D,E
FE57    1A              DATAOT: LDAX    D               ;LD A INDIRECT FROM D & E
FE58    CDD0FE                  CALL    OUTPUT          ;OUTPUT TO TAPE
FE5B    13                      INX     D               ;BUMP POINTER
FE5C    81                      ADD     C               ;ADD TO CHECK SUM
FE5D    4F                      MOV     C,A             ;REPLACE IN CKSUM
FE5E    05                      DCR     B               ;DECREMENT BYTE COUNT
FE5F    C257FE                  JNZ     DATAOT          ;NOT ZERO, WRITE ANOTHER
                        ;
                        ;       WRITE CHECK SUM
                        ;
FE62    79                      MOV     A,C             ;CKSM IN A
FE63    CDD0FE                  CALL    OUTPUT          ;OUTPUT IT TO TAPE
FE66    2F                      CMA
FE67    CDD0FE                  CALL    OUTPUT
                        ;
FE6A    21FEFE                  LXI     H,DBC           ;DATA BYTE COUNT
FE6D    7E                      MOV     A,M             ;IN A REG
```

```
FE6E    FEFF                CPI     255             ;IF LT, WRITE END OF FILE
FE70    C288FE              JNZ     EOF
FE73    2AFCFE              LHLD    LPRN            ;SAME FILE #, NEW RECORD#
FE76    23                  INX     H
FE77    22FCFE              SHLD    LPRN            ;PUT NEW RECORD NUMBER BACK IN BUF
FE7A    110001              LXI     D,256           ;THIS SECTION BUMPS SA BY 256
FE7D    00                  NOP
FE7E    2A00F8              LHLD    LSA             ;TO SEC3 FOR NEXT
FE81    19                  DAD     D               ;RECORD OUTPUT
FE82    2200F8              SHLD    LSA
FE85    C30CFE              JMP     SEC3
                    ;
                    ;       WRITE FILE MARK AND FILE NUMBER
                    ;
FE88    CDB7FE      EOF:    CALL    TMOT            ;PRODUCES INTER RECORD GAP
FE8B    3E81                MVI     A,81H           ;FILE MARK CODE
FE8D    3201FF              STA     BUF+2
FE90    2F                  CMA
FE91    3202FF              STA     BUF+3           ;SETS UP BUFFER FOR FILE I.D. OUTPUT
FE94    0606                MVI     B,6             ;SET INDEX
FE96    CDC3FE              CALL    BUFOUT          ;OUTPUT BUFFER TO TAPE
FE99    CDB7FE              CALL    TMOT            ;PRODUCES INTER RECORD GAP
                    ;
                    ;       SEC 4 WRITES END OF FILE ON TAPE
                    ;
FE9C    00          SEC4:   NOP
FE9D    3EFF                MVI     A,0FFH          ;END OF FILE CODE
FE9F    3203FF              STA     BUF+4
FEA2    2F                  CMA
FEA3    3204FF              STA     BUF+5           ;SETS UP FOR EOF MARK
FEA6    0606                MVI     B,6             ;SET UP INDEX
FEA8    CDC3FE              CALL    BUFOUT          ;OUTPUT EOF TO TAPE
FEAB    3EFE                MVI     A,0FEH          ;RETURN TO INITIAL CONDS.
FEAD    3203FF              STA     BUF+4
FEB0    2F                  CMA
FEB1    3204FF              STA     BUF+5
FEB4    C3B4FE      FINISH: JMP     FINISH          ;TO EXIT, PUSH STOP
                    ;
FEB7    1EFF        TMOT:   MVI     E,0FFH          ;MAX NUMBER
FEB9    53          LP1:    MOV     D,E
FEBA    15          LP2:    DCR     D
FEBB    C2BAFE              JNZ     LP2
FEBE    1D                  DCR     E
FEBF    C2B9FE              JNZ     LP1
FEC2    C9                  RET
FEC3    21FFFE      BUFOUT: LXI     H,BUF           ;SET MEM POINTER TO BUFFER
FEC6    7E          LOOP:   MOV     A,M             ;MOVE BUFFER(N) TO A
FEC7    CDD0FE              CALL    OUTPUT          ;OUTPUT TO TAPE
FECA    23                  INX     H               ;BUMP MEM POINTER
FECB    05                  DCR     B               ;DECREMENT INDEX
FECC    C2C6FE              JNZ     LOOP            ;DO AGAIN IF INDEX NOT ZERO
FECF    C9                  RET                     ;RETURN
                    ;
FED0    F5          OUTPUT: PUSH    PSW             ;STORE A ON STACK
FED1    DB6E        LP:     IN      CASC            ;INPUT CASSETTE STATUS
FED3    E620                ANI     20H             ;CLEAR ALL BUT STATUS BIT
FED5    C2D1FE              JNZ     LP              ;CHECK AGAIN UNTIL ZERO
FED8    F1                  POP     PSW             ;RELOAD A REG FROM STACK
FED9    D36F                OUT     CASD            ;OUTPUT IT TO CASSETTE
FEDB    C9                  RET                     ;RETURN
                    ;
FEDC    2A00F8      RECLGN: LHLD    LSA
FEDF    EB                  XCHG                    ;START ADDRESS IN D & E
FEE0    2A02F8              LHLD    LST             ;STOP ADDRESS IN H & L
FEE3    7B                  MOV     A,E             ;SET UP TWO'S COMP
FEE4    2F                  CMA                     ;OF START ADDRESS, THEN
FEE5    5F                  MOV     E,A             ;ADD TO STOP ADDRESS TO
FEE6    7A                  MOV     A,D             ;SEE IF WITHIN 255 RANGE.
FEE7    2F                  CMA
FEE8    57                  MOV     D,A
FEE9    13                  INX     D               ;TWO'S COMP
FEEA    19                  DAD     D               ;H & L CONTAIN DIFF BETWEEN S/A &ST/A
FEEB    3E00                MVI     A,0             ;ZERO A REG
FEED    BC                  CMP     H               ;SETS CONDITION CODES
FEEE    C2F5FE              JNZ     GT
FEF1    7D                  MOV     A,L             ;SIZE OF RECORD NOW IN A
FEF2    C3F7FE              JMP     BC              ;JUMP TO DATAOUT
FEF5    3EFF        GT:     MVI     A,255           ;MAX BYTE COUNT
FEF7    32FEFE      BC:     STA     DBC             ;STORE IN DATA BYTE COUNT
FEFA    C9                  RET                     ;A CONTAINS BYTECOUNT
0001                FILEN:  DS      1
0001                LPRN:   DS      1
0001                MPRN:   DS      1
0001                DBC:    DS      1
FEFF    3CE6817E    BUF:    DW      0E63CH,7E81H,1FEH,0,0,0,0,0
FF03    FE010000
FF07    00000000
FF0B    00000000
```

Gary Kay
Southwest Technical Products Corp.
219 W. Rhapsody
San Antonio TX 78216



SWTPC's new 8K memory board.

Expand Your SWTP 6800

... with a new 8K board

Until now it has been impossible to expand the SWTPC 6800 Computer System to its 32K memory limit without going to the extreme of adding another mainframe and power supply. The mainframe has provisions for up to six memory boards and with 4K boards, a maximum of only 24K may be implemented on a single mainframe. On top of this SWTPC recommends a maximum of four MP-M 4K memory boards on a single mainframe due to the 1.5 Amp per board power consumption on a 10 Amp maximum 5 volt power supply. Now there is an 8K static memory board available for the system. The board uses the new 18-pin 4K static memories being offered by Texas Instruments, National, Mostek and other manufacturers. These static memories, like the well-known 2102, offer the simplicity and reliability not always achieved with dynamic memories.

The memory chips are fully static requiring no clocks or refresh circuitry. They are organized 4096 x 1 with three-state outputs and chip select control for OR-tie capability (also called dot-OR, wire-OR, etc.) They are fully TTL compatible and operate on a single 5 volt power supply. Average power dissipation per IC varies from 220 mW to 325 mW depending upon the manufacturer.

Sixteen of these 4K memory chips along with the address decode/buffers, data bus transceivers and voltage regulator ICs have been implemented on the 5½" x 9" double sided, plated-through hole, MP-8M memory board. All connections from the memory board to the SS-50 standard mother board are made through the 50-pin Molex edge connector soldered to the lower edge of the MP-8M. The PC board layout on the MP-8M has been arranged so that all of the regulators, decoders and buffers are in a parallel row along the lower edge of the circuit board close to the connector pins where they belong. The two 4K increments of memory are directly above in two parallel rows with the upper row for the upper 4K of memory and the lower row for the lower 4K of memory. Horizontal chip arrangement is bit 7 to bit 0 going from left to right, which greatly simplifies PC board tracing and debugging should it ever be necessary.

The memory board is provided with a write/protect switch which can be flipped to the PROTECT position to prevent the memory contents of the entire board from being changed until the switch is returned to the NORMAL position or the system is powered down. Address selection for the board is done through an onboard DIP switch rather than jumpers. Each board may be configured for 0 – 8K, 8K – 16K, 16K – 24K or 24K – 32K operation simply by setting the address selection switch.

The most outstanding feature of the board has to be the ease with which it may be assembled. On the bottom (soldered side) of the circuit board there are no lines running between pads of the ICs and there are no lines narrower than 0.031". Assembly of the board should be no more difficult than the other boards on the computer system. The top side of the circuit board does have lines running between pads on some of the ICs and there are some 0.020" wide lines, but the foil is widely spaced. Chances for solder and foil bridges and breaks are considerably reduced over that of the original MP-M 4K memory board.

Total power consumption for the circuit board will vary with different manufacturer's memories but worst case for the entire 8K board should be no more than 1.5 Amps which is that specified for the MP-M 4K memory board. Thus the MP-8M 8K board gives twice the memory as the MP-M 4K board for the same power consumption.

The cycle time required for the memories used with the SWTPC 6800 Computer System is 550 ns or less. This is an easy specification for the 4K static memories since production yields are peaking well under this figure.

How it Works

Each of the memory chips on the board is a 4096 x 1-bit static RAM IC storing one of the eight bits of each word

within one of the two 4,096 word memory halves. The actual bit and memory assignment of each memory IC is given in Table 1. The twelve address lines A_0 through A_{11}, as well as the R/W lines of all of the memory ICs on each board are paralleled. Because of the large capacitances generated by this paralleling, integrated circuits IC19 and IC20 are used as noninverting buffers to drive these lines. The actual address selection for each of the two eight-bit 4,096 word halves is done using integrated circuits IC18, IC21, and IC22.

Since the eight-bit data bus for the computer system is bidirectional, bidirectional transceiver/buffers IC16 and IC17 buffer the incoming and outgoing data to and from the memory board to the data bus. Integrated circuits IC21 and IC22 enable the outgoing sections of the bidirectional transceivers at the appropriate times while the incoming sections are enabled at all times since the memory ICs have separate input/output lines. The +5 volt power for the decoders and buffers (IC16-IC22) is supplied by voltage regulator IC23 while +5 volt power for the memories is supplied by voltage regulator IC24. The various capacitors are used to reduce power supply bus noise.

The MP-8M board is sold in kit form only by Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio, Texas 78216. The kit includes the circuit board, all components and assembly instructions and sells for $250 ppd in the U.S. ■

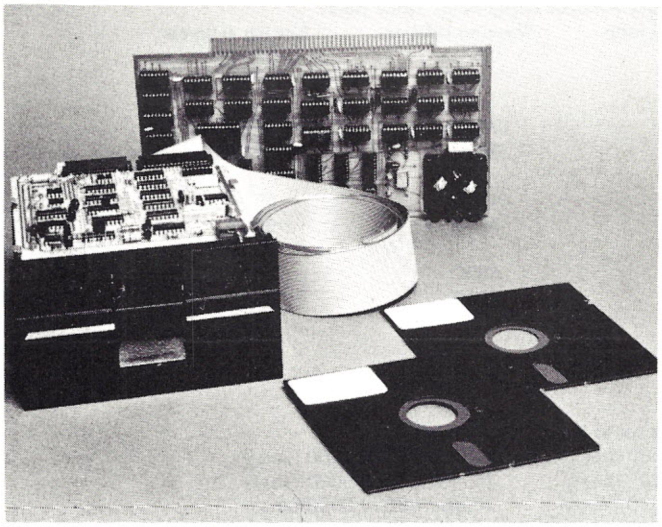
MP-8M Memory Address Assignment Table (hex)

Board Select	Half of Memory	Starting Address	Ending Address
8	lower	0000	0FFF
	upper	1000	1FFF
16	lower	2000	2FFF
	upper	3000	3FFF
24	lower	4000	4FFF
	upper	5000	5FFF
32	lower	6000	6FFF
	upper	7000	7FFF

Hex to Binary Conversion

00 hex = 0000 0000 binary	08 hex = 0000 1000 binary
01 hex = 0000 0001 binary	10 hex = 0001 0000 binary
02 hex = 0000 0010 binary	20 hex = 0010 0000 binary
04 hex = 0000 0100 binary	40 hex = 0100 0000 binary
	80 hex = 1000 0000 binary

Table 1. MP-8M Memory Address Assignments (hex). Hex to Binary Conversion.



COMPLETE FLOPPY DISK SYSTEM FOR YOUR ALTAIR/IMSAI $699

That's right, complete.

The North Star MICRO-DISK SYSTEM™ uses the Shugart minifloppy™ disk drive. The controller is an S-100 compatible PC board with on-board PROM for bootstrap load. It can control up to three drives, either with or without interrupts. No DMA is required.

No system is complete without software: we provide the PROM bootstrap, a file-oriented disk operating system (2k bytes), and our powerful extended BASIC with sequential and random disk file accessing (10k bytes).

Each 5" diameter diskette has 90k data byte capacity. BASIC loads in less than 2 seconds. The drive itself can be mounted **inside** your computer, and use your **existing** power supply (.9 amp at 5V and 1.6 amp at 12V max). Or, if you prefer, we offer a power supply ($39) and enclosure ($39).

Sound unbelievable? See the North Star MICRO-DISK SYSTEM at your local computer store. For a high-performance BASIC computing system, all you need is an 8080 or Z80 computer, 16k of memory, a terminal, and the North Star MICRO-DISK SYSTEM. For additional performance, obtain up to a factor of ten increase in BASIC execution speed by also ordering the North Star hardware Floating Point Board (FPB-A). Use of the FPB-A also saves about 1k of memory by eliminating software arithmetic routines.

Included: North Star controller kit (highest quality PC board and components, sockets for all IC's, and power regulation for one drive), SA-400 drive (assembled and tested), cabling and connectors, 2 diskettes (one containing file DOS and BASIC), complete hardware and software documentation, and U.S. shipping.

MICRO-DISK SYSTEM . . . $699
(ASSEMBLED) $799
ADDITIONAL DRIVES. . . $425 ea.
DISKETTES. . . . . $4.50 ea.
FPB-A $359
(ASSEMBLED) $499

To place order, send check, money order or BA or MC card # with exp. date and signature. Uncertified checks require 6 weeks processing. Calif. residents add sales tax.

N9

NORTH STAR COMPUTERS, INC.
2465 Fourth Street
Berkeley, CA 94710

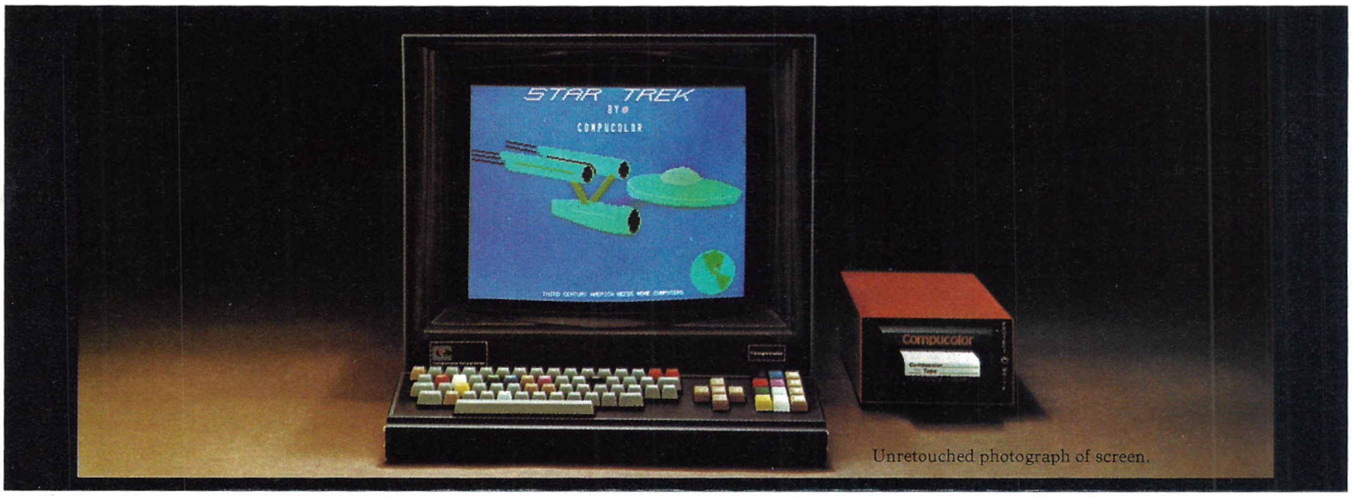
BUY YOUR COMPUCOLOR 8001 FROM THESE DEALERS.

ARIZONA			
Phoenix:	Phoenix Byte Shop West 12654 N. 28th Dr.	(602) 942-7300
Tempe:	Tempe Byte Shop East 813 N. Scottsdale Rd.	(602) 894-1129
Tucson:	Byte Shop of Tucson 2612 E. Broadway	(602) 327-4579
CALIFORNIA			
Lawndale:	Byte Shop of Lawndale 16508 Hawthorne Blvd.	(213) 371-2421
Orange:	Computer Mart of Los Angeles 625 W. Katella, No. 10	(714) 633-1222
San Diego:	The Computer Center 8205 Ronson Rd.	(714) 292-5302
San Francisco:	The Computer Store 1093 Mission St.	(415) 431-0640
San Jose:	Amco Electronics 414 S. Bascom Ave.	(408) 998-2828
San Jose:	Byte Shop 155 Blossom Hill Rd.	(408) 226-8383
Van Nuys:	Computer Components 5848 Sepulveda Blvd.	(213) 786-7411
CONNECTICUT			
Windsor Locks:	The Computer Store 63 S. Main St.	(203) 627-0188
FLORIDA			
Coral Gables:	Sunny Computer Stores 1238A S. Dixie Hwy.	(305) 661-6042
Tampa:	Microcomputer Systems 144 S. Dale Mabry Hwy.	(813) 879-4301
GEORGIA			
Atlanta:	Atlanta Computer Mart 5091-B Buford Hwy.	(404) 455-0647
HAWAII			
Honolulu:	Compact Computers P.O. Box 10096	(808) 373-2751
ILLINOIS			
Champaign:	The Numbers Racket 518 E. Green St.	(217) 352-5435
Evanston:	Itty Bitty Machine 1316 Chicago Ave.	(312) 328-6800
Park Ridge:	Chicago Computer Store 157 Talcott Rd., Hwy. 62	(312) 823-2388
INDIANA			
Indianapolis:	Home Computer Shop 10447 Chris Dr.	(317) 894-3319
MARYLAND			
Rockville:	Computer Workshop, Inc. 5709 Frederick Ave.	(301) 468-0455
MASSACHUSETTS			
Burlington:	The Computer Store 120 Cambridge St.	(617) 272-8770
MICHIGAN			
Troy:	General Computer Company 2017 Livernois	(313) 362-0022
MINNESOTA			
Minneapolis:	Cost Reduction Services 3142 Hennepin Ave. So.	(612) 822-2119
MISSOURI			
Kansas City:	Computer Workshop of Kansas City 6903 Blair Rd.	(816) 741-5055
NEW JERSEY			
Iselin:	The Computer Mart 501 Route 27	(201) 283-0600
NEW YORK			
East Meadow:	The Computer Mart of Long Island 2070 Front St.	(516) 794-0510
New York:	The Computer Mart 314 5th Ave.	(212) 279-1048
OHIO			
Columbus:	Computervision 894 W. Broad St.	(614) 228-2477
SOUTH CAROLINA			
Columbia:	Byte Shop 2018 Greene St.	(803) 771-7824
TEXAS			
Houston:	Communications Center 7231 Fondren	(713) 774-9526
Richardson:	The Micro Store 634 S. Central Expressway	(214) 231-1096
WASHINGTON			
Seattle:	Retail Computer Store 410 NE 72nd St.	(206) 524-4101
WISCONSIN			
Beloit:	Austin Computers 1835 Northgate	(608) 365-6096
Watertown:	General Precision Electronics 207 Rhine St.	(414) 261-8148

Compucolor® Corporation

16

NOW $2750. AMERICA'S LOWEST-PRICED PERSONAL COMPUTER SYSTEM WITH COLOR VECTOR GRAPHICS.



Unretouched photograph of screen.

By taking advantage of the new technologies available to the industry today, we've consistently been able to give you one of the best prices on the market. Now because of great response, we can give you the best price. You can now buy the Compucolor 8001 for the reduced price of $2750. A complete stand-alone system with expanded graphics software for plotting points, vectors and bargraphs on a 160 x 192 addressable grid—in color. Eight independent background and foreground colors.

The Compucolor 8001 has an Intel 8080 CPU, 34 I/O ports and a color display with an effective band width of 75 MHZ compared to 5 MHZ for standard TV sets. In fact the Compucolor is the only totally integrated system on the market which includes a color display. You can also have special options for the Compucolor 8001 right now, including: Mini Disk Drives for extra memory, light pens and a variety of special keyboard features.

BASIC 8001 Is Easy To Learn. Compucolor's BASIC 8001 is a conversational programming language which uses English-type statements and familiar mathematical notations. It's simple to learn and easy to use, too. Especially when it comes to intricate manipulations or expressing problems more efficiently. The BASIC 8001 Interpreter runs in ROM memory and includes 26 statement types, 18 mathematical functions, 9 string functions and 7 command types for executing, loading, saving, erasing, continuing, clearing or listing the program currently in core.

Expandable Memory To 64K. The Compucolor 8001 has 11K bytes of non-destructible read-only memory which handles the CPU and CRT operating systems as well as BASIC 8001. Sockets are in place for an additional 21K of EPROM/MROM memory. The Random Access Main Memory has 8K bytes for screen refresh and scratch pad, 8K bytes for user workspace and room for 16K bytes of additional user workspace. The Compucolor also comes complete with a convenient mass storage device, Floppy Tape Memory. It's an 8-track continuous loop tape system, with a Baud rate of 4800 and an extra storage capacity of up to 1024K bytes per tape.

Color Graphics At Alphanumeric Black And White Prices.

That's what we're becoming famous for, and thanks to the tremendous response to the Compucolor 8001, we've been able to reduce our price even lower—to $2750. Look over our dealer listing on the adjacent page for the dealer nearest you. Then drop by for a demonstration. And while you're checking out the Compucolor 8001, check out your dealer's financing plan. He can help you turn a good deal into a good deal more.

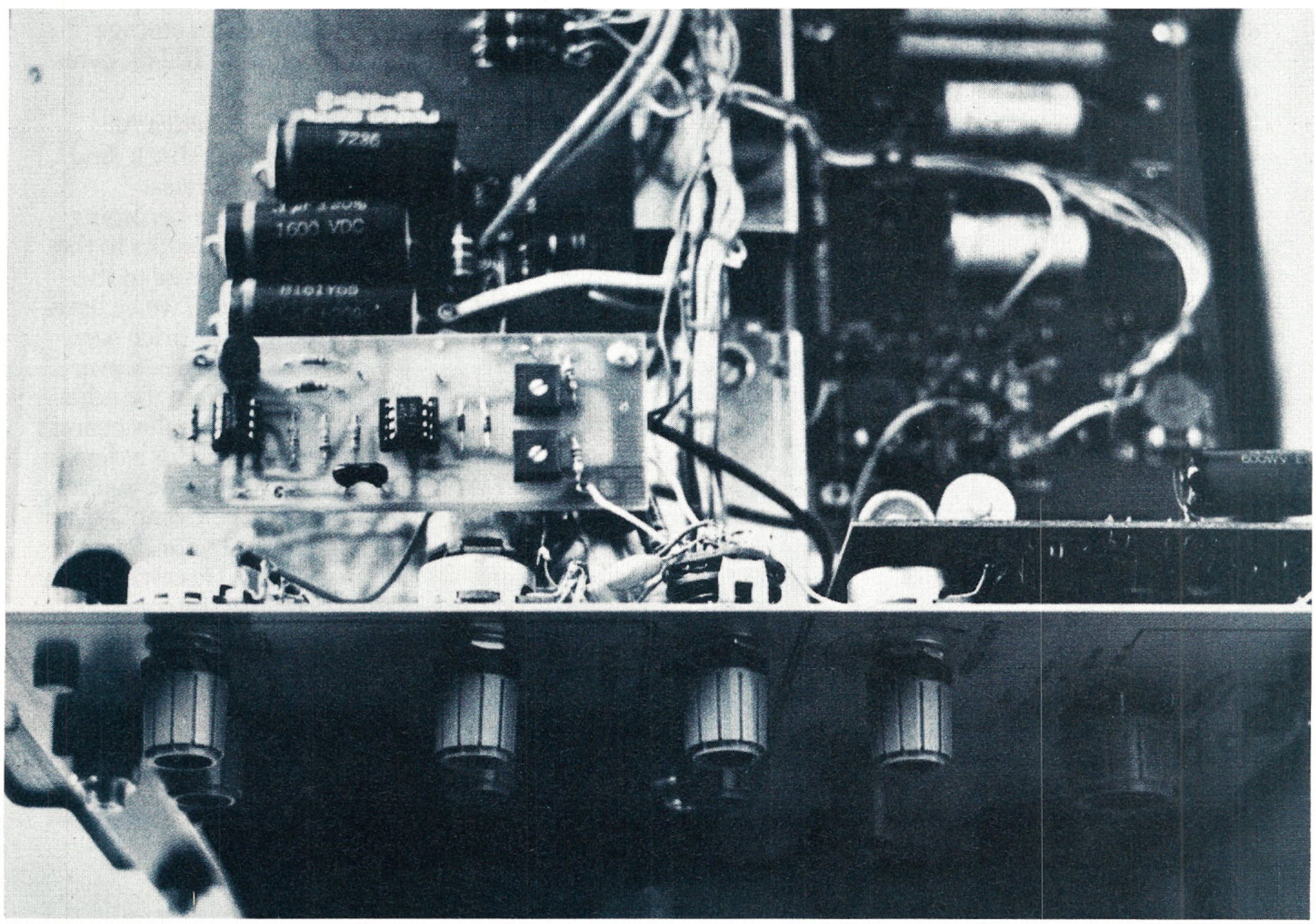
Compucolor Corporation, P.O. Box 569, Norcross, Georgia 30091.

Compucolor Corporation

Trigger Your Oscilloscope

. . . add value to inexpensive 'scopes

Sweep circuit board mounted in the Heathkit IO-102 scope and the additional panel controls (Time Base and Frequency Vernier) located below the position controls. Knobs were purchased from Heath to match the original panel appearance.



In Bill's article "Build an Eight Channel Multiplexer for Your Scope" (Issue #4) we promised a follow- up article describing the construction of a triggered sweep circuit for untriggered scopes. Here it is. Having a triggered sweep for troubleshooting digital circuits is the only way to go. I've seen people using untriggered scopes for such troubleshooting . . . but they sure didn't have any idea when events were taking place.

Bill also provides us with some excellent material on oscilloscope theory in this article (and there are a lot of us who use scopes . . . but don't understand them all that well). — John.

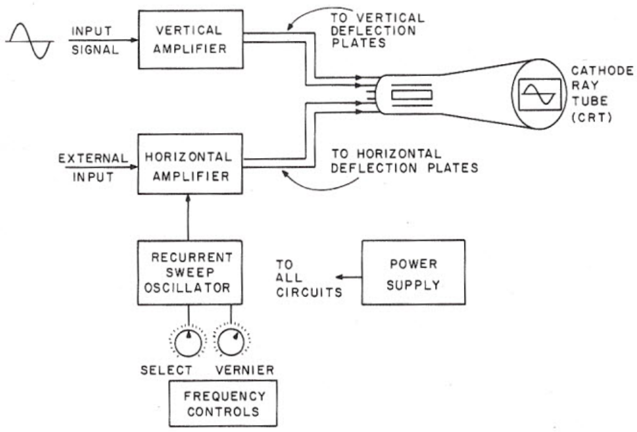


Fig. 1. Block diagram of typical general purpose oscilloscope with recurrent type horizontal sweep. This type of horizontal oscillator is "free-running" and is not necessarily in phase with the input signal. The resultant display will sometimes tend to drift across the screen.

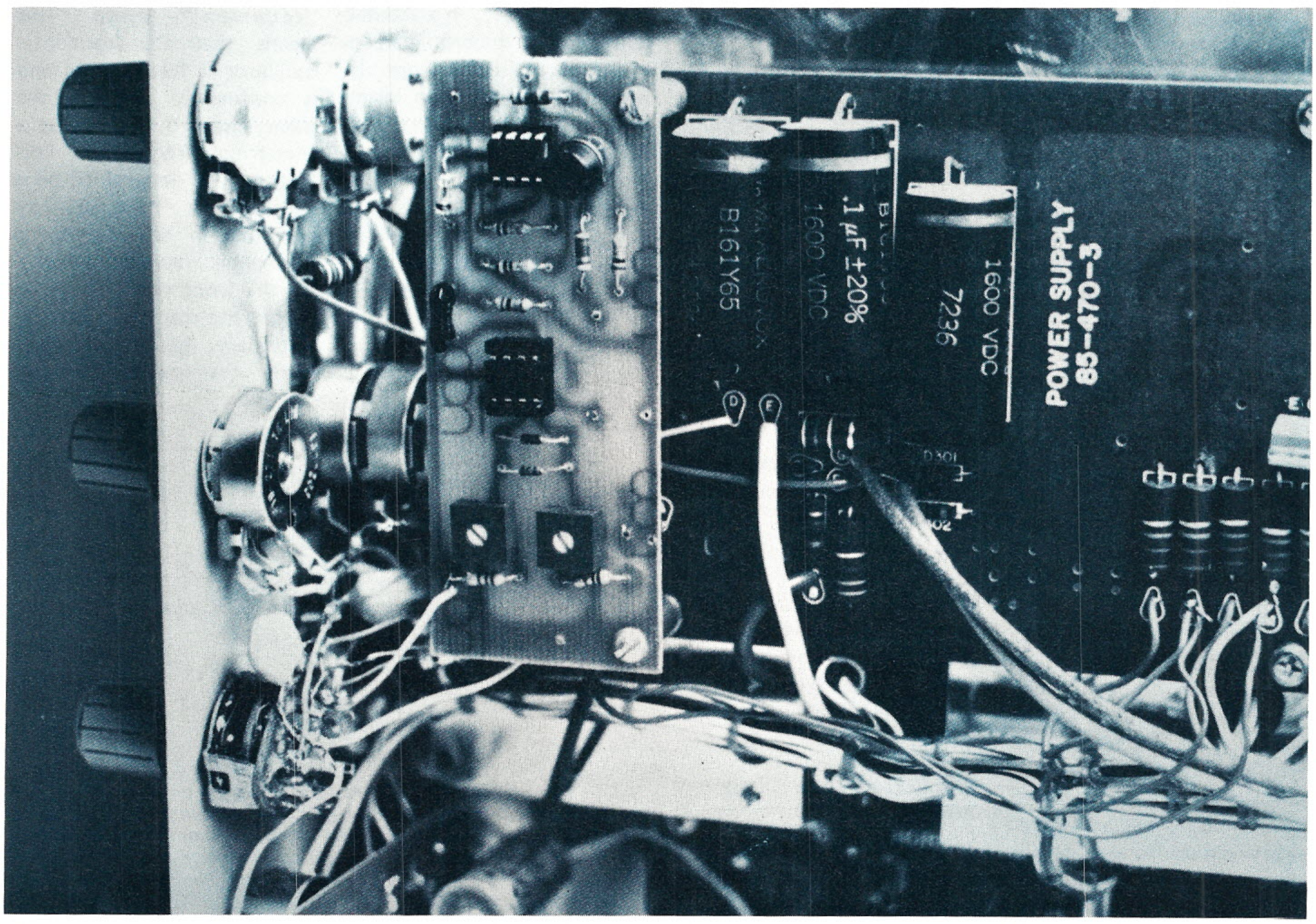
W. J. Prudhomme
1405 Richland Ave.
Metairie LA 70001

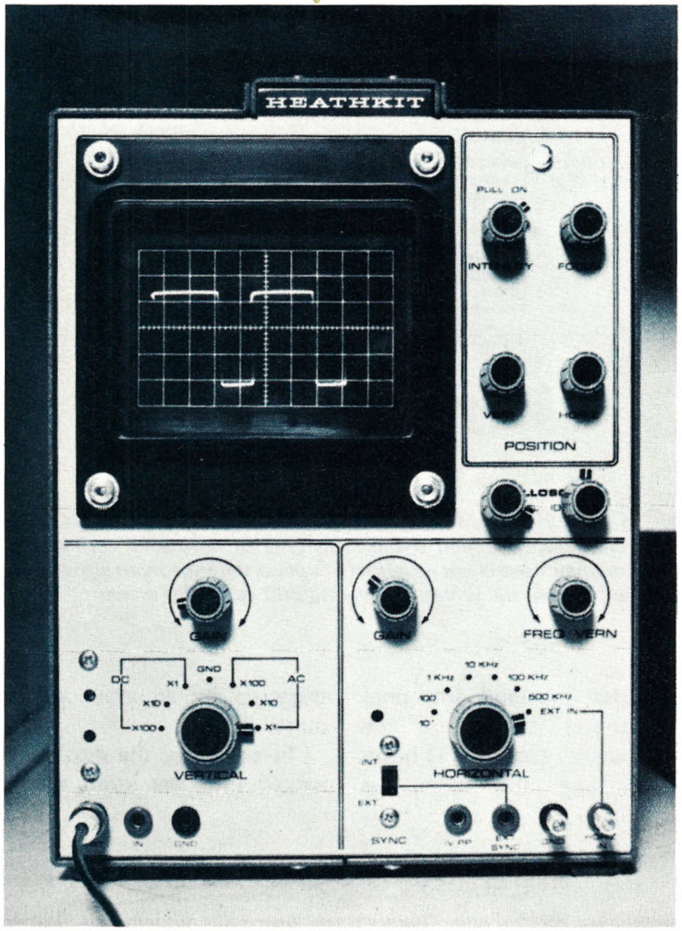
The oscilloscope is perhaps one of the most useful test instruments available to us in troubleshooting digital circuits. Not only does it display logic levels in the circuit being tested, a scope also allows us to "stop" the action and compare the timing relationships between various high speed digital pulses throughout the circuit.

To accomplish this feat, most lab grade oscilloscopes feature a "triggered sweep." Basically, this is a circuit that provides one (and only one) horizontal sweep of the cathode ray tube (CRT) beam each time the circuit is triggered by an input signal pulse.

In so doing, the display is locked into the input signal

The triggered sweep circuit board may be mounted in a convenient space within the oscilloscope. This particular board was mounted on the low voltage power supply of a Heathkit IO-102 scope.





Rock stable display is possible with a triggered sweep. New controls are just below the original position controls. The two added knobs operate the Time Base selection and the Frequency Vernier.

and this results in a rock stable display on the CRT screen. Just how the triggered sweep circuit accomplishes this will be discussed in more detail later. The point is, a stable scope display is essential in troubleshooting digital circuits and this is best accomplished by a scope with a triggered sweep.

Since a lab grade oscilloscope with a triggered sweep may cost as much as a kilobuck, most hobbyists are reluctant to invest this much money in a test instrument when there are so many other items more urgently needed. Also, what if you already own a general purpose oscilloscope without a triggered sweep? Well, hopefully this article will help solve that problem, since it describes how to easily add a triggered sweep circuit to any oscilloscope. The best part is that it can be done for just a few dollars and a couple of hours of your time.

The circuit requires only two low cost ICs and a transistor. You also have the option of installing the circuit internal to your scope (with slight modifications) or outboard with no changes required of the scope.

Before describing the actual construction of an add-on triggered sweep, some background information on how oscilloscopes function may be useful in understanding the internal circuitry. Those of you who are already familiar with scope circuits may want to skip the next section and go on to the actual construction details.

Recurrent vs. Triggered Sweep

All oscilloscopes contain the following basic elements: vertical and horizontal amplifiers, horizontal sweep circuit, cathode ray tube (CRT) for display, and a power supply. Referring to the block diagram in Fig. 1, let's examine each individual circuit and its primary function.

First, the signal to be displayed is coupled into the vertical amplifier where it is amplified hundreds of times and coupled to two plates within the CRT. These plates are obviously called the vertical deflection plates since they cause the electron beam in the tube to be deflected in a vertical direction in proportion to the input signal. Naturally, the amount of input attentuation and the gain-setting of the vertical amplifier both determine how much the beam will be deflected. At this point, if nothing else acted on the beam, all we would see on the CRT screen would be a vertical line and the oscilloscope would be useful only as an indicator of voltage levels.

To make the oscilloscope more useful, horizontal deflection of the beam can be added to our basic scope. If we sweep the electron beam in the CRT in a horizontal direction at the proper speed and in unison with the vertical signal, we get a composite picture of what the input signal really looks like with respect to time.

In order to accomplish this horizontal sweep of the beam, it's necessary to add a horizontal oscillator, horizontal amplifier and two horizontal deflection plates within the CRT. To keep the display looking clean, we must add a blanking circuit to the horizontal circuitry.

The reason for the blanking circuit becomes clear when we examine the movement of the electron beam. Initially, the beam starts on the extreme left-hand side of the screen and moves to the extreme right-hand side. This completes one horizontal sweep and the beam must be returned to the left side of the screen to begin another sweep. When this occurs, the blanking circuit turns the beam off while it is returning to the left side of the screen. This eliminates extra traces on the screen that would be confused with the actual signal trace.

The problem with most general purpose oscilloscopes lies in the horizontal oscillator circuit. These scopes use what is referred to as a "recurrent" sweep. This means that the horizontal oscillator is free-running and is continually sweeping the trace across the screen at a recurring frequency. This frequency is determined by a resistor-capacitor combination selected by a horizontal frequency control on the front panel. In order to get the display to remain stationary, it's necessary to trim the oscillator's fre-

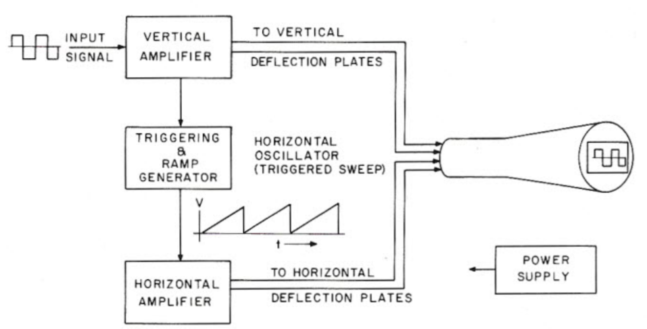


Fig. 2. Block diagram of an oscilloscope with a triggered sweep. The input signal is also coupled to a trigger circuit which provides a pulse to the ramp generator when the input signal exceeds a preset level. The horizontal oscillator then provides a ramp voltage and is immune to further triggering until it completes one cycle. The resultant display is stable and always in "sync."

quency with a vernier control (also on the front panel) until it matches the frequency of the input signal. Only when these two frequencies are closely matched will they tend to "lock in" and give a stationary display. The actual adjustment in many instances can be tricky if not downright frustrating.

Also, if the input signal is varying in frequency, it will tend to drift away from the initial setting of the horizontal frequency control. In this case, it is virtually impossible to keep the display stationary for any period of time.

To overcome these problems the triggered sweep circuit was developed, which adds a new dimension to the usefulness of the oscilloscope. A block diagram of this type of scope is shown in Fig. 2. Here the vertical input (in addition to being connected to the vertical amplifier) is also connected to a second amplifier and triggering circuit. The purpose of this circuit is to provide a trigger pulse to a ramp generator when the input signal reaches a certain predetermined level. The triggering level can be selected by controlling the gain of the trigger amplifier. The trigger pulse causes a one-shot multivibrator to charge an RC combination resulting in the generation of a ramp voltage for the horizontal amplifier. Note that only one ramp generated voltage occurs per trigger pulse, and the duration of the ramp voltage is determined by the RC combination.

Once the one-shot multivibrator has been triggered, it is immune to further triggering for the duration of its cycle. Hence, only one horizontal sweep occurs per trigger even though the input signal may be going through several cycles during the horizontal sweep. This results in an extremely stable display since the horizontal sweep is always triggered and locked into the input signal. By selecting different time bases for the horizontal ramp voltage, it's also possible to obtain any number of input signal cycles and always have a stable display.

That's the basic difference between recurrent and triggered sweep circuits and why the latter is so much more superior. The question is how do you add a triggered sweep to your scope if it lacks one? Thanks to modern IC technology, the complexity of such a circuit has been greatly simplified and a practical circuit is described in the next section.

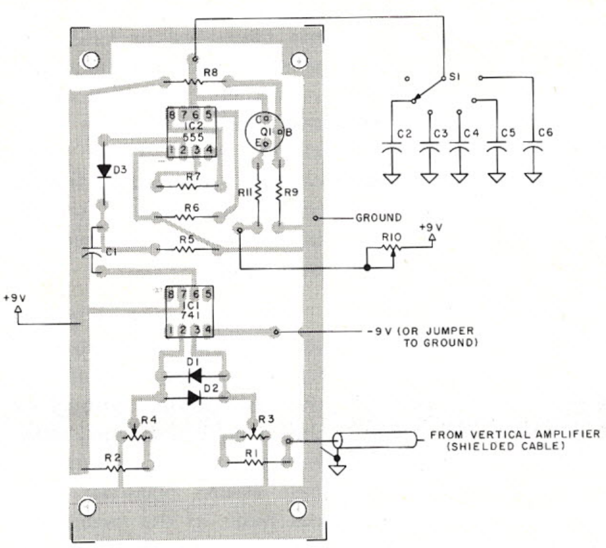


Fig. 4b. Component placement and external connections.

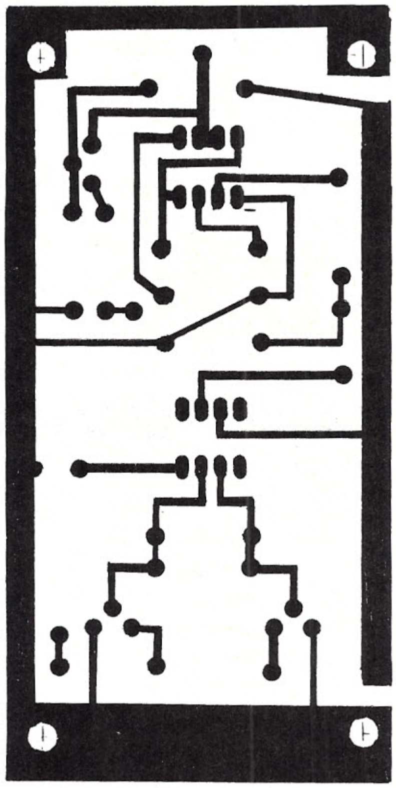


Fig. 4a. Printed circuit board layout (full size).

Circuit Description

The circuit diagram for the add-on triggered sweep is shown in Fig. 3. For the amplifier portion, a 741 op amp is used with the noninverting input connected to the scope's vertical amplifier and the inverting input used to control the trigger level. Normally, the output of the 741 is at the +V level. When an input signal rises above the trigger level setting, the output of the op amp swings to -V which is capacitively coupled to the negative trigger input of a 555 timer. This negative spike causes the output of the timer to go "high" allowing the output capacitor to charge through a transistor and resistor in series. The effect of the transistor is to provide constant current charging of the capacitor resulting in a nearly perfect ramp voltage.

When the capacitor charges to a level equal to 2/3 V, the 555 timer resets itself and waits for the next trigger pulse. In this manner, the frequency of the ramp voltage is determined by the RC combination and may be varied from 1 Hz to 1 MHz.

The output ramp voltage is then coupled to the scope horizontal amplifier to provide a linear horizontal sweep triggered by the vertical input signal. The circuit may be constructed in an external enclosure and connected to the input terminals of the scope, or it may be mounted internal to the scope deriving its power from the scope's internal power supply. The next section describes the actual construction and installation of the sweep circuit.

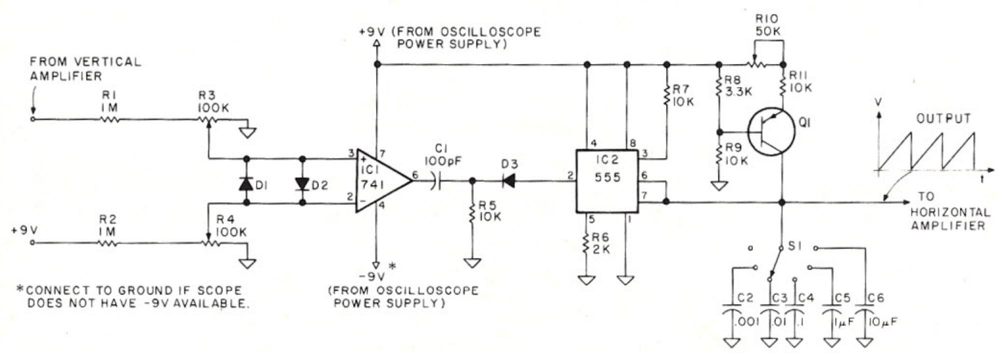


Fig. 3. Schematic diagram.

C1 — 100 pF disk, 25 volt
C2 — 0.001, 25 V
C3 — 0.01, 25 V
C4 — 0.1, 25 V
C5 — 1 uF, 25 V
C6 — 10 uF, 25 V
(C2–C6: All Polyester (Mylar) or Tantalum)
D1, D2, D3 — 1N914 diodes
IC1 — 741 op amp (DIP)
IC2 — 555 timer
R1, R2 — 1 meg
R3, R4 — 100k pot
R5, R7, R9, R11 — 10k
R6 — 2k
R8 — 3.3k
R10 — 50k linear taper pot
S1 — 1 pole, 6 pos. rotary switch
All fixed resistors ¼ Watt, 10%
R3, R4 — PC board type trim pots
Q1 may be any PNP switching type transistor.

Parts list.

Construction and Installation

The triggered sweep circuit may be constructed on a 4" x 2" PC board, and a typical PC layout is shown in Fig. 4 along with the placement of parts. Layout is not particularly critical, and you may want to rearrange the board size to accommodate space available in your own scope. The board described here was designed to be installed in a Heathkit IO-102 scope as shown in the accompanying photographs, but it's applicable to other scopes as well.

I chose to install the circuit inside the scope, while retaining the integrity of the existing recurrent sweep circuit. The Horizontal Frequency Selector on the IO-102 has a position for external sweep. When the selector is in this position, the internal horizontal sweep oscillator is disabled and the horizontal amplifier input is connected to external input jack.

This feature turned out to be ideal for my purposes since it gives the option of recurrent or triggered sweep modes. The circuit board was mounted above the low voltage power supply circuit board near the horizontal amplifier. Both +9 volts and -9 volts are available in the IO-102. The voltages available in other scopes may be different and you may have to add a regulator circuit depending on the voltage levels you encounter.

Two controls were added to the front panel of the scope just below the horizontal and vertical position controls. These were the frequency or time base selector and the frequency vernier controls as shown in the photographs. The sensitivity and trigger level controls are mounted directly on the circuit board and are set at the level that gives the best results.

The input of the triggered sweep circuit was connected to the vertical amplifier of the oscilloscope through a short length of shielded cable. For the IO-102, I found the best results occurred when the input of the sweep amp was connected to the collector of the vertical output transistor (Q9 on the Heathkit schematic). However, you may want to experiment with several points in the vertical amplifier of your own scope until you find the best connection point. Beginning with the initial input stage, each successive amplifier stage should provide a high voltage level with maximum signal available at the output stage. Since the characteristics of each brand of scope may be different, you will have to experiment until you get the best results with the least interference with the operation of the vertical amplifier.

If you choose to construct the triggered sweep circuit in an external enclosure, connections may be made to the vertical and horizontal inputs on the scope's front panel. However, at low input signal levels, there may be insufficient output from the 741 to trigger the 555. To overcome this problem, it may be necessary to add another amplifier stage, possibly with an FET transistor (featuring high input impedance) or another op amp stage (using a dual 741).

Regardless of which option you select, the addition of a triggered sweep to your scope should greatly enhance its usefulness in troubleshooting. In fact, once you get used to the stable displays, you'll wonder how you ever got along without a triggered sweep. ■

References

Berlin, H. M., *The 555 Timer Applications Sourcebook,* E & L Instruments, Inc.
Anderson, R. L., "555 Timer Sweep Circuit for SSTV," *73 Magazine* May, 1976 pg. 134.

CORRECTIONS

I would like to correct a small error that appeared in D. LaDage's article "Interrupts Exposed," April, 1977. In discussing interrupt handling by the 8080, Dan states that the RST instruction disables interrupts while the CALL instruction does not. This is not true. After receiving an interrupt, the 8080 automatically (by pulling INTE off) disables interrupts. It does this regardless of the instruction jammed onto the bus by the interrupting device. The only way to enable interrupts is to execute the EI instruction (this sets INTE). Thus, one need not worry about being interrupted while in an interrupt service routine if you do want to be since an EI instruction must be executed before any further interrupts can be recognized.

By executing an EI instruction upon entering an interrupt servicing routine, "nested" interrupts are possible; however, status saving and stack size sometimes make programming for "nested" interrupts tricky. Regardless, do not forget to execute an EI instruction before leaving an interrupt service routine if you want interrupts enabled.

Intel has recently announced the 8259 Programmable Interrupt Controller which makes interrupt handling extremely flexible. The 8259 manages 8 levels of interrupts and has built-in features allowing expandability to 64 levels with the addition of other 8259's. All levels can be prioritized by a selection of programmalbe priority modes. Separate CALLS for each level can be placed anywhere in the address map. All modes can be changed under software control allowing the complete interrupt structure to be dynamically defined as required, based on the system environment. Consult the Intel Microcomputer Peripheral User's Manual for details on the 8259.

John Beaston
Intel Corporation
Santa Clara CA

In "Inside the Sphere Microcontroller" (July), Fig. 6 on page 24 should have the following changes: additions under the column heading "Sphere" – line 1, "See above"; line 2, "0000"; line 3, "See above"; line 4, "6000"; line 8, "DFCC"; line 9, "DF 96". Under the column heading "Operation" line 9 should read "Output 2 hex char." In Program A (page 25) line 00060 should be ---CHARACTER CONVERTS---; line 00082 should be — ' # —; (page 26) line 00091 should be — IND REG(X) —; line 00165 should be — (and stop —; (page 27) the comment between lines 00179 and 00180 should instead occur between lines 00180 and 00182; (page 28) line 00283 should add IL in the fifth column; line 00328 should be — 2D —.

In "News of the Industry" (July, page 18), column 1, the fifth and sixth lines from the bottom should be changed to read as follows: "FORTRAN-80 may be purchased for $500 . . ."

Al Gerbens
1038 E. 6th Place
Mesa AZ 85203

Sobriety Tester Program

... logic conquers Demon Rum!

There are two ways you might view the following article . . . in a serious vein or as another computer game. I personally prefer the serious approach — for two reasons. First, I think we've been laughing at drunkards too long and it's time to stop taking it so lightly. Second, if Al's program saves just one life (because someone decided to take a cab or be driven home), then it's possibly one of the most significant pieces of software we'll ever publish. Like anything else, it can probably do with some improvements here and there. If you have any, please send them in. We'd also be interested in hearing of your experiences with the program, okay? — John.

Here's a program that will make your microcomputer the center of attention at your next cocktail party. Imagine your pride and joy being admired by your friends and no one asking "What can you use it for?" for the entire evening.

What's it do? It first asks you a few questions about yourself, then tests your memory retention, mental concentration and physical reaction time. The program subsequently directs your computer to make a few calculations and output a number between 0 and 100, that number corresponding to the degree of your intoxication. That sounds simple enough; here's how it works.

A person's degree of sobriety is a function of many factors, including the following which are considered in the first part of the program: 1) Alcohol intake during the past few hours. 2) Tolerance level of the individual. 3) Whether or not a meal has been eaten in the past few hours. 4) The individual's body weight representing a dilution factor.

The program weights the number of ounces (approximate) consumed in the past 4 hours with factors of

- latest hour (30)
- first hour before (26)
- second hour before (22)
- third hour before (18)

You may disagree with these weighting factors; they are quite subjective and if you would like to change them look at line 206. D(1); D(2); D(3) and D(4) are the number of drinks consumed in the past 4 hours. Just change the multipliers to change the weighting.

Back on line 206 notice that the next variable is E and it is subtracted. E represents the response to the "have you eaten . . ." question. If the answer is yes, the E becomes 30 units; if no, then E = 0 units. V then is building up to represent increased intoxication and being subtracted from to represent increased sobriety. The variable 5 represents tolerance, and 4 times your 0 to 10 input is subtracted from V. One fifth of the body weight entry L is then subtracted to yield V.

V therefore 1) becomes larger as alcohol intake increases; 2) is reduced by 30 if a heavy meal has been consumed in the last 3 hours; 3) is reduced by a factor between 0 and 40 depending upon an individual's estimate of his own tolerance level; 4) is reduced by one fifth of the individual's body weight in pounds.

Look at line 208. The value of Q is added to the V number which was calculated in line 206. Q is the total time in arbitrary units accumulated during the reaction time part of the test. This sequence begins at line 184 and continues through line 204. The time before the appearance of the next number is determined by the number of blank lines (# "") generated after the correct key is hit. To keep this time from being consistent, the number of blank lines generated is between 16 and 32 as a function of a random number generated in line 188.

```
                    RUN



                    ==============
                *                !
             *        DEMON      !
   --------
   !                             !
   !                             !
  0 --------                     !
 000            *     RUM        !
0000               *             !
0000                ==============
000000
00000000
0000000000
000000000000
00000000000000
0000000000000000
000000000000000000
00000000000000000000
0000000000000000000000
00000000000          0000000000
00000000000 SOBRIETY 0000000000
00000000000          0000000000
00000000000 TEST     0000000000
00000000000          0000000000
0000000000000000000000000000000
0000000000000000000000000000000
0000000000000000000000000000000
0000000000000000000000000000000
0000000000000000000000000000000


HOW MANY DRINKS HAVE YOU HAD IN
THE LAST HOUR?
(ASSUME 1 BEER = 1 DRINK) 2
HOW MANY DRINKS IN THE THREE
HOURS PRIOR TO THIS LAST HOUR?
HOUR BEFORE LAST — 1
THE HOUR BEFORE THAT? 0
AND THE HOUR BEFORE THAT 0
THANK YOU

HAVE YOU EATEN A HEAVY MEAL IN
THE LAST 3 HOURS?
ENTER A Y OR N N

ON A SCALE OF 0 TO 10 WITH 0
REPRESENTING A TEE-TOTALER AND
10 REPRESENTING A CONFIRMED LUSH
HOW DO YOU RATE YOURSELF?
? 5

WHAT IS YOUR WEIGHT IN LBS? 200
THANK YOU
WE WILL CONTINUE WITH A
PERFORMANCE EVALUATION.
THIS IS A MEMORY RETENTION
PART OF THE TEST.
A NUMBER WILL BE WRITTEN AT
THE BOTTOM OF THE SCREEN AND
MOVE UP TO THE MIDDLE OF THE
SCREEN WHERE IT WILL PAUSE
FOR A SHORT PERIOD OF TIME.
REMEMBER THE DIGITS IN ORDER
AND REENTER THEM WHEN ASKED.
WHEN ENTERING THE NUMBERS
TYPE A SINGLE DIGIT AT A TIME
AND PUSH THE RETURN KEY AFTER
EACH DIGIT.
HERE THEY COME
1 3 7 9 2 0

PLEASE TYPE THE SIX DIGITS YOU
JUST OBSERVED
? 1
? 3
? 7
? 9
? 2
? 0
THIS SEQUENCE REPEATS
4 MORE TIMES.
THIS IS THE REACTIVE TIME PART
OF THIS EXAM.
A NUMBER FROM 0 TO 9 WILL
APPEAR ON THE SCREEN.
PRESS THE KEY WITH THAT NUMBER
ON IT AS QUICKLY AS YOU CAN
GOOD LUCK

6
(PRESS 6)
1
(PRESS 1)
THIS SEQUENCE REPEATS FOR
A TOTAL OF 10 TIMES.
ON A SCALE OF 0 TO 100 WITH
0 BEING STONE SOBER AND 100
BEING BOMBED: YOUR SCORE IS
16
ANYONE ELSE? N
READY
```

Sample Run.

Line 196 uses an Inp (0) function. The keyboard on my system uses parallel input port 0. With no key depressed the computer sees 127_{10}. The sequence from line 196 through line 202 is therefore a scan input port Ø sequence with each execution of the loop incrementing the value of Q by 1. When the computer recognizes the number it is displaying, the loop is terminated and the process of generating the next number begins. When it's all said and done, Q is a number proportional to the total time it took you to recognize a single digit number appearing on the screen and then to push the key with that number on it for each of the 10 numbers displayed. Theoretically Q should become larger as the degree of intoxication becomes greater. Therefore, Q is simply added to V in line 208. In line 210 the number of correct entries (T) in the memory retention portion of the test is divided by 30 (the total number of digits displayed) and this is divided into the V+Q value. This will increase the total score as the number of correct entries decreases. The total V(2) is then divided by 40 in line 214 to scale it between 0 and 100. It's possible to generate a score greater than 100, but not very probable. If you top 100, then you automatically get a score of 110. Anyone getting a score of 110 should have pillows placed around the chair he is sitting on.

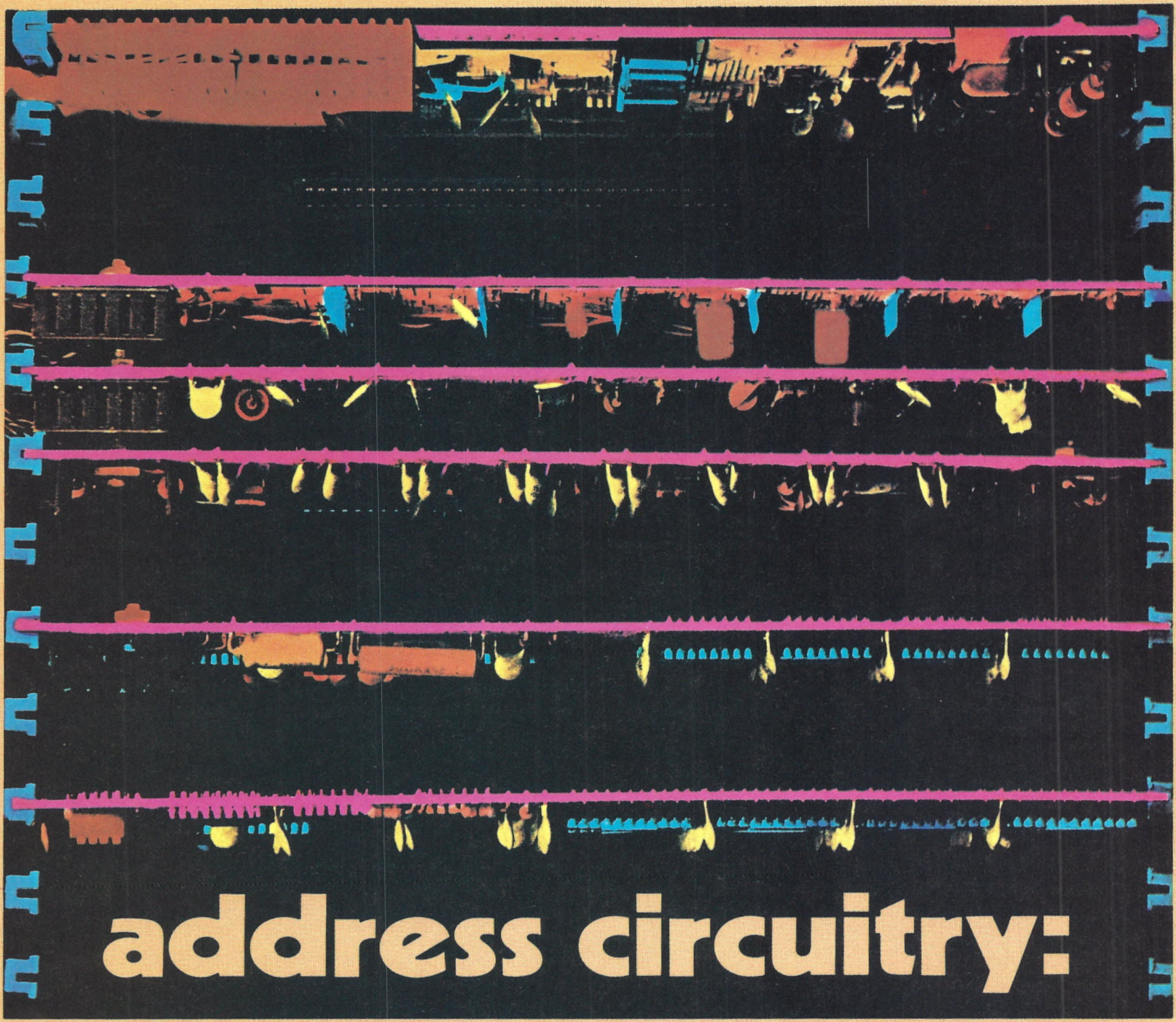
At this point very little data has been accumulated on the validity of the output number. I would be interested in learning of your experience with the program and suggestions relative to scaling.

The program is written in Digital Group Maxi-BASIC version one, and I have it running on a Digital Group Z80 18K system. If the BASIC you're using doesn't have an "Inp" function you might consider using the "call" or "user" functions and writing a machine language subroutine to generate the display and timing loop. If you're short on memory, the first thing to delete is the "picture of a bottle" routine. Each of the three sections could also be executed independently.

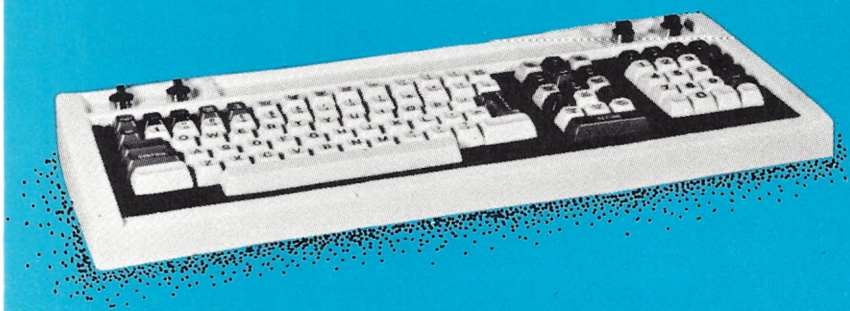
The variety of memory retention tests which could be incorporated is fascinating. How about a short paragraph of information followed by questions relating to the subject of the paragraph, or pattern recognition and retention using a graphics display? ■

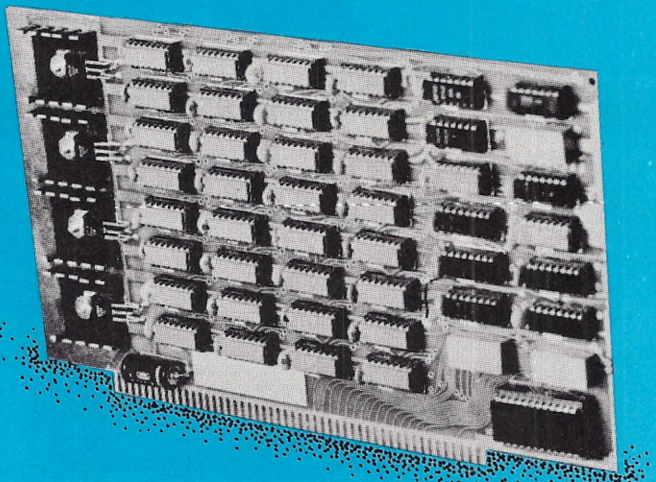
```
10 GOSUB 266
12 #"HOW MANY DRINKS HAVE YOU HAD IN"
14 #"THE LAST HOUR?"
16 INPUT "(ASSUME 1 BEER = 1 DRINK)", D(1)
18 IF D(1) > 8 THEN #"ENTRY TOO LARGE, RE-ENTER"
20 IF D(1) > 8 THEN 16
22 #"HOW MANY DRINKS IN THE THREE"
24 #"HOURS PRIOR TO THIS LAST HOUR?"
26 INPUT "HOUR BEFORE LAST - ", D(2)
28 INPUT "THE HOUR BEFORE THAT?",D(3)
30 INPUT "AND THE HOUR BEFORE THAT", D(4)
32 IF D(2)+D(3)+D(4) >24 THEN #"UNBELIEVABLE! REENTER"
34 IF D(2)+D(3)+D(4) >24 THEN 26
36 #"THANK YOU": FOR X=1 TO 500: NEXT
38 FOR X=1 TO 16: #"": NEXT
40 #"HAVE YOU EATEN A HEAVY MEAL IN"
42 #"THE LAST 3 HOURS?"
44 INPUT "ENTER Y OR N", A$
46 IF A$="N" THEN E=0
48 IF A$="Y" THEN E=30
50 FOR X=1 TO 16: #"": NEXT
52 #"ON A SCALE OF 0 TO 10 WITH 0"
54 #"REPRESENTING A TEE-TOTALER AND"
56 #"10 REPRESENTING A CONFIRMED LUSH"
58 #"HOW DO YOU RATE YOURSELF?"
60 INPUT S
62 IF S >10 THEN 58
64 IF S <0 THEN 58
66 FOR X=1 TO 16: #"": NEXT
68 INPUT "WHAT IS YOUR WEIGHT IN LBS?",L
70 IF L <50 THEN 68
72 IF L >300 then 68
74 FOR X=1 TO 16: #"": NEXT
76 #"THANK YOU, WE WILL CONTINUE"
78 #"WITH A PERFORMANCE EVALUATION."
80 FOR X=1 TO 8: #"": NEXT
82 FOR X=1 TO 1000: NEXT
84 FOR X=1 TO 18: #"": NEXT
86 #"THIS IS A MEMORY RETENTION"
88 #"PART OF THE TEST.": FOR X=1 TO 2000: NEXT
90 #"A NUMBER WILL BE WRITTEN AT"
92 #"THE BOTTOM OF THE SCREEN AND"
94 #"MOVE UP TO THE MIDDLE OF"
96 #"THE SCREEN WHERE IT WILL PAUSE"
98 #"FOR A SHORT PERIOD OF TIME."
100 #"REMEMBER THE DIGITS IN ORDER"
102 #"AND REENTER THEM WHEN ASKED."
104 FOR X=1 TO 5000: NEXT
106 #""
108 #"WHEN ENTERING THE NUMBERS"
110 #"TYPE A SINGLE DIGIT AT A TIME"
112 #"AND PUSH THE RETURN KEY AFTER"
114 #"EACH DIGIT."
116 FOR X=1 TO 5000: NEXT
118 #"HERE THEY COME": FOR X=1 TO 20: NEXT
120 Q=1000
122 FOR G=1 TO 5
124 FOR X=1 TO 32: #"": NEXT
126 #TAB (8);
128 FOR W=1 TO 6
130 N(W)=INT (9*RND(0))
132 #N(W);
134 NEXT
136 FOR Y=1 TO 8:#"": NEXT
138 FOR Z=1 TO Q: NEXT
140 FOR X=1 TO 9: #"": NEXT
142 #"PLEASE TYPE THE 6 DIGITS YOU JUST OBSERVED"
144 FOR W=1 TO 6
146 INPUT M(W)
148 IF M(W) >9 THEN #"INVALID ENTRY REENTER"
150 IF M(W) <Ø THEN #" INVALID ENTRY REENTER"
152 NEXT
154 #"THANK YOU"
156 FOR W=1 TO 6
158 IF N(W)=M(W) THEN T=T=1
160 NEXT
162 FOR X=1 TO 16: #"": NEXT
164 NEXT G
166 #" THIS IS THE REACTION TIME PART"
168 #"OF THIS EXAM." FOR X=1 TO 1500: NEXT
170 #"A NUMBER FROM 0 TO 9 WILL"
172 #"APPEAR ON THE SCREEN."
174 FOR X=1 TO 2000: NEXT
176 #"PRESS THE KEY WITH THAT NUMBER"
178 #" ON IT AS QUICKLY AS YOU CAN.": FOR X=1 TO 2000: NEXT
180 #"GOOD LUCK!": FOR X=1 TO 1000: NEXT
182 FOR X=1 TO 20: #"": NEXT
184 Q=0
186 FOR H=1 TO 10
188 FOR X=1 TO (16+16*RND(0)): #"": NEXT
190 D=INT (9*RND(0))
192 P=176+D
194 #D
196 I=INP(0)
198 Q=Q+1
200 IF I=127 THEN 196
202 IF P <> I THEN 196
204 NEXT H
206 V=30 *D(1)+26*D(2)+22*D(3)+18*D(4)—E—4*S—L/5
208 V(1)=V+Q
210 V(2)=V(1)/(T/30)
212 IF D(1)+D(2)+D(3)+D(4)=0 THEN 232
214 V(3)=V(2)/40
216 IF V(3) >100 THEN V(3)=110
218 #"ON A SCALE OF 0 TO 100 WITH"
220 #"0 BEING STONE SOBER AND 100"
222 #"BEING BOMBED: YOUR SCORE IS"
224 #INT (V(3))
226 INPUT "ANYONE ELSE?", A$
228 IF A$ "Y" THEN 12
230 END
232 IF Q/(T/30) <800 THEN 236
234 IF Q/(T/30) >800 THEN 246
236 #"YOU REPORT HAVING CONSUMED"
238 #"NO ALCOHOL IN THE PAST FOUR HOURS."
240 #"YOUR MENTAL RETENTION AND PHY-"
242 #"SICAL REACTIONS APPEAR NORMAL."
244 GO TO 230
246 #"DESPITE THE FACT YOU REPORT 0"
248 #"ALCOHOL INTAKE, YOUR MENTAL"
250 #"RETENTION AND PHYSICAL REACTIONS"
252 #"ARE POOR. FATIGUE, ILLNESS,"
254 #"INFLUENCE OF OTHER DRUGS AND"
256 #"MEDICINES CAN BE CONTRIBUTING"
258 #"FACTORS."
260 #"RECOMMEND YOU CORRECT BEFORE"
262 #"DRIVING."
264 GO TO 230
266 FOR X=1 TO 16: #"":NEXT
268 #TAB(14);"= = = = = = = = = = = = = ="
270 #TAB(13);"*                              !"
272 "TAB(12);"* DEMON                        !"
274 #"       - - - - - - - - ";
276 #"TAB (28);"!"
278 FOR X=1 TO 2: #"  !";
280 #TAB (23); "X";
282 #"    !":NEXT
284 #" 0- - - - - - - - -            !"
286 #" 000      *  RUM        !"
288 #"0000       *          !"
290 #"0000       = = = = = = = = = = = = = = = "
292 FOR X=1 TO 3000:NEXT
294 FOR X=6 TO 28 STEP 2
296 X1=X
298 #"0";
300 X1=X1-1:IF X1 <> Ø THEN 298
302 #"0"
304 NEXT
306 #"00000000000                 0000000000"
308 #"00000000000 SOBRIETY 0000000000"
310 #"00000000000                 0000000000"
312 #"00000000000 TEST            0000000000"
314 #"00000000000                 0000000000"
316 FOR X=1 TO 5
318 #"00000000000000000000000000000000":NEXT
320 FOR X=1 TO 1000:NEXT
322 FOR X=1 TO 16: #"":NEXT
324 RETURN
```

Program Listing.


address circuitry:

NEW FROM.....





DS-STAND ALONE KEYBOARD

- ☐ Full 102 CHERRY key switches.
- ☐ Output serial or parallel. Includes cursor pad, hex pad and user defined keys.
- ☐ Output is switch selectable for upper or lower case.
- ☐ N-Key roll over and depression repeat. Internal power supply. 110/60 cy.
- ☐ Strong ABS cabinet. 21"x7". 8 lbs.
- ☐ L.E.D. lights indicate output.
- ☐ Baud rates up to 19,260.
- ☐ All functions switch selectable from keyboard.

$149.95 KIT / $225 ASSEMBLED

DS-16K RAM MEMORY BOARD

- ☐ Uses fast 350NS 4k X 1 rams.
- ☐ Expandable to 64K using 16K rams. Transparent memory refresh—no wait.
- ☐ Each 4K segment addressable to any 4K boundary. (Dip switches)
- ☐ Fully plated through and solder masked P.C. board.
- ☐ Low-profile sockets for all chips.
- ☐ All boards tested for 180 hours
- ☐ Fully S-100 bus compatible.

FULLY ASSEMBLED AND TESTED
$298

DS-STAND ALONE INTERFACE CARD

- ☐ Unique micro-interface card allows a DS-Stand alone owner to interface his keyboard directly to an S-100 bus computer without an addition interface card.
- ☐ May be set for any port, parity, etc.

$29.95 KIT / $49.95 ASSEMBLED

DS - S/P/S ADAPTER

- ☐ Serial/parallel/serial Universal interface adapter.
- ☐ Completely self-contained serial to parallel to serial converter in a single package.
- ☐ Switch selectable baud rates up to 19,260.
- ☐ Complete with ABS cabinet, 120/60 cy.

$69.95 KIT / $99.95 ASSEMBLED

Random Integer Program

... for games, sorting, or statistics

Philip Tubb
ALF Products Inc
128 South Taft
Denver CO 80228

For many computer games, it is necessary to have a program pick a set of integers randomly, but with no integer occurring more than once. A typical example would be any game that deals cards. An integer from 1 to 52 can be used to represent each card. Each time a new card must be dealt it is necessary to pick a random integer from 1 to 52 which has not been picked (dealt) already. In BASIC the obvious method is to construct a loop to pick the numbers and store them in an array. Each time a number is picked the array can be scanned to make sure it has not already been picked (see Example 1).

The problem with this approach is that it may take quite a long time, especially on picking the 52nd number.

```
10 DIM A(52)
20 FOR A=1 TO 52
30 B=INT(52*RND(0))+1
40 FOR C=1 TO A-1
50 IF A(C)=B THEN 30
60 NEXT C
70 A(A)=B
80 NEXT A
```

Example 1

```
        LXI   H,TABLE+51
LOOP1   MOV   M.L          INITIALIZE THE DECK.
        DCR   L
        JNZ   LOOP1
        MVI   B,H
        MVI   L,52
LOOP2   CALL  RANDOM       PICK NUMBER BETWEEN 1 AND (L).
        MOV   C,A          MOVE RANDOM VALUE INTO C.
        LDAX  B            LOAD PICKED CARD VALUE.
        MOV   D,A          SAVE IN D.
        MOV   A,M          LOAD "LAST" CARD.
        STAX  B            STORE AT PICKED LOCATION.
        MOV   M,D          STORE PICKED CARD AT LAST LOCATION.
        DCR   L
        JNZ   LOOP2        LOOP UNTIL DONE.
```

Example 5

A simple solution to this can be observed from actually dealing a deck of cards. On the first card, one out of 52 choices is dealt and removed from the deck. On the second, one out of 51 is picked and removed. This continues to the last card. This can be simulated in BASIC as shown in Example 2.

```
10 DIM A(52),B(52)
20 FOR A=1 TO 52
30 A(A)=A
40 NEXT A
50 FOR A=1 TO 52
60 B=INT((53-A)*RND(0))+1
70 B(A)=A(B)
80 FOR C=B+1 TO 53-A
90 A(C-1)=A(C)
100 NEXT C
110 NEXT A
```

Example 2

Lines 20 through 40 set up the deck in array A. Lines 50 through 110 deal the cards into array B. Lines 80 through 100 remove a card from array A after it has been dealt. Although this procedure will work and will be much faster than the first example, it is not necessary to move the entire array down each time a card is dealt. The object of moving the array down is to put the last card into the array where it can be picked and to remove the card which was picked so it will not be picked again. Since it doesn't matter what order the cards are in (in the original array) a simpler method would be to copy the last card in the array into the element of the card which was picked (see Example 3).

```
10 DIM A(52),B(52)
20 FOR A=1 TO 52
30 A(A)=A
40 NEXT A
50 FOR A=52 TO 1 STEP -1
60 B=INT(A*RND(0))+1
70 B(53-A)=A(B)
80 A(B)=A(A)
90 NEXT A
```

Example 3

Again, 20 through 40 set up the deck. 50 through 90 deal the cards into array B. Line 80 moves the *last* card into the element of the just picked card. One problem with this program is that it deals the cards from one array into another. This requires two arrays. If the cards could be simply rearranged in the array, only one array would be required. This can be done by swapping the last card with the picked card as shown in Example 4.

```
10 DIM A(52)
20 FOR A=1 TO 52
30 A(A)=A
40 NEXT A
50 FOR A=52 TO 2 STEP -1
60 B=INT(A*RND(0))+1
70 C=A(A)
80 A(A)=A(B)
90 A(B)=C
100 NEXT A
```

Example 4

The cards can be dealt from the A array as needed simply by keeping a variable which indicates the next element to use. Initialize it to one and increment it each time a card is dealt.

In 8080 assembly language the program is relatively simple. Assuming a routine named RANDOM exists which picks a random integer from 1 to a value specified by the L register. This can be done either with a software random number generator such as XOR and shift (Example 6) or divide, add, and use the remainder; or with a hardware random number generator such as the ALF Products 10-5-5. In either case the random number is divided by the contents of L and the remainder is used as the random number after adding one to bring the range from between zero and (L)-1 to one and (L). The random integers are placed in memory at the address specified by TABLE. This address must be (for simplicity in the program) such that the low byte is one. See Example 5.

The cards are dealt by maintaining a pointer into memory. It is initialized to TABLE and incremented each time a card is dealt. ■

The Small Computer

Twenty-five years ago a computer as powerful as the new Processor Technology Sol-20 priced out at a cool million.

Now for only $995 in kit form or $1495 fully assembled and tested you can have your own small computer with perhaps even more power. It comes in a package about the size of a typewriter. And there's nothing like it on the market today. Not from IBM, Burroughs, DEC, HP or anybody else!

It fills a new role

If you're an engineer, scientist or businessman, the Sol-20 can help you solve many or all of your design problems, help you quantify research, and handle the books too. For not much more than the price of a good calculator, you can have high level computer power.

Use it in the office, lab, plant or home

Sol-20 is a smart terminal for distributed processing. Sol-20 is a stand alone computer for data collection, handling and analysis. Sol-20 is a text editor. In fact, Sol-20 is the key element of a full fledged computer system including hardware, software and peripheral gear. It's a computer system with a keyboard, extra memory, I/O interfaces, factory backup, service notes, users group.

It's a computer you can take home after hours to play or create sophisticated games, do your personal books and taxes, and a whole host of other tasks.

Those of you who are familiar with small computers will recognize what an advance the Sol-20 is.

Sol-20 offers all these features as standard:

8080 microprocessor — 1024 character video display circuitry — control PROM memory — 1024 words of static low-power RAM — 1024 words of preprogrammed PROM — built-in cassette interface capable of controlling two recorders at 1200 bits per second — both parallel and serial standardized interface connectors — a complete power supply including ultra quiet fan — a beautiful case with solid walnut sides — software which includes a preprogrammed PROM personality module and a data cassette with BASIC-5 language plus two sophisticated computer video games — the ability to work with all S-100 bus products.

Full expansion capability

Tailor the Sol-20 system to your applications with our complete line of peripheral products. These include the video monitor, audio cassette and digital tape systems, dual floppy disc system, expansion memories, and interfaces.

Write for our new 22 page catalog. Get all the details.

Processor Technology, Box Y 6200 Hollis St., Emeryville, CA 94608. (415) 652-8080.

Processor Technology Corporation

Ron Drafz
916 So. Grant
Villa Park IL 60181

Test ICs With Your Micro

... the micro as a valuable test instrument

Normally we prefer to go with actual working circuits instead of articles which are "think" pieces. Due to the lack of some detail it appears Ron's article is such a think piece. It sure is neat and splendid in all its simplicity, though! I've been looking for something on an IC tester (using a micro) for a long time. If someone wants to take Ron's scheme, try it out, and then write it up in a full-blown article with photos, detailed diagrams, flowcharts, software and whatever . . . I think a lot of people would be interested in reading about it. — John.

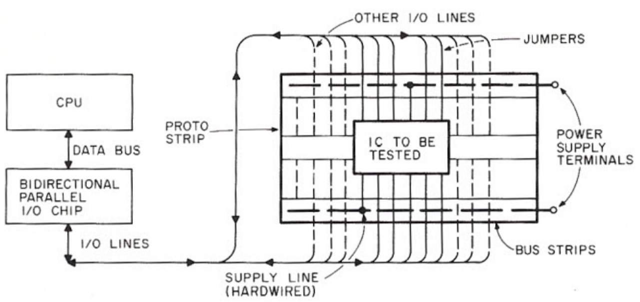


Fig. 1. Typical IC test system using proto-strip.

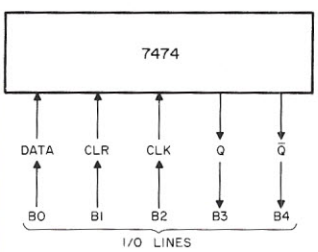


Fig. 2. Test set-up for 7474 flip-flop.

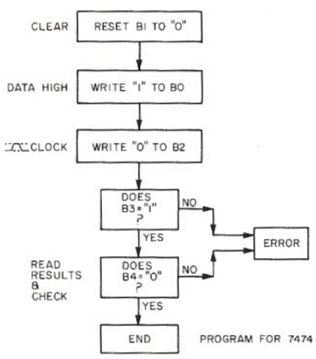


Fig. 3. Flowchart for 7474 test program.

K*ilobaud* . . . Finally, a microcomputer magazine dedicated solely to hobbyists. And speaking of hobbyists, I would like to share an application to make the computer a useful test instrument. One interesting use for a microcomputer is to check integrated circuits such as memories on its own PC cards. The computer itself can be used to check virtually any type of logic gate with the proper interfacing, making it an ideal troubleshooting instrument.

The easiest way to implement this idea is to use the microprocessor's input/output (I/O) capability. Bidirectional, or two-way, I/O ports are extremely useful because the user only has to worry about the software to set up the proper logic states for the I/O lines. The I/O chips that work best are the 8255 Parallel I/O chip for the Z-80/8080 CPU, the 6820 PIA for the MC6800 microprocessor, or the standard set of bidirectional I/O lines of the Fairchild F-8 chip set. A typical setup is shown in Fig. 1. A proto-strip is needed for mounting the ICs.

Jumpers are needed that connect directly from the I/O ports to the proto-strip, which are simply pushed in. However, since a typical TTL IC's power supply lines draw more current than the I/O port can handle, the I/O port jumpers should be disconnected from these pins, and wires substituted leading from the pins to the adjoining bus strips carrying supply voltages.

Using this type of configuration, virtually any kind of logic IC can be tested including RAMs, ROMs, or CMOS as long as they are TTL compatible (+5 V levels for the I/O ports). The microprocessor can be used to create clock pulses for a particular IC, such as the example of the 7474 shown in Fig. 2. The microprocessor sets up clock pulses by writing 1s and 0s alternately on a specified I/O pin after a certain delay period. The results of the IC's action is accomplished by reading the output pin and testing it for the proper information (see flowchart, Fig. 3).

RAM memories can be checked by writing data into them, then reading the data out and checking it. EPROMs can be verified for the correct codes. Programmable ROMs of all types can be checked for all zeroes before programming.

The real beauty of this type of IC logic tester is that it is not limited to a specific type of integrated circuit. The only requirement for the IC is that it is dual-in-line and TTL compatible. You simply plug in the IC and call the appropriate subroutine program to test it. The possibilities are unlimited! ■

Heavy Duty Altair Power Supply

... plenty of power for peripherals

Rudy not only presents a super construction article for building an Altair-bus power supply but he also discusses some power supply fundamentals in a way that will be of interest to the newcomer . . . and serve as a refresher for those who have been away from power supplies for awhile. And wait until you see the price of this little beast! — John.

Dr. Rudolf Hirschmann
1001 Kagawa St.
Pacific Palisades CA 90272

Do you still have one of the original Altair 8800s? If so, you know what a power crisis can mean. Or are you building microcomputer equipment of your own? If so, then you will need a suitable power source, and in either case this article can help. It will show how to upgrade the power supply of the original Altair so that it can run a full complement of plug-in boards and still have power to spare. Everything will fit in the Altair case and cost $60 or less.

The heart of this supply is a new version of a Southern California Computer Society (SCCS) — designed power transformer that is intended to replace all three transformers of the original Altair. Many members of SCCS, however, realized that it doesn't *have* to be used in an Altair. It can also become the heart of a new power supply, and in this capacity it can simultaneously provide — with direct fan cooling — 8V at 20 A as well as plus and minus 16 V at 2 A each. With no cooling at all you will have to derate these currents by 40%. These figures are the ones suggested by the manufacturer, and my measurements show them to

Height in inches	Capacitance in uF	Price in $	Manufacturer	Model number
4.125	95,000	8.60	Mallory	CG953U015X4C
4.25	100,000	13.30	Cornell Dubillier	FAH100000-15-D3
4.25	120,000	22.70	General Electric	86F525
4.125	130,000	13.00	Mallory	CGS134UO15X4C
5.625	140,000	11.45	Mallory	CG144UO15X5L
5.625	150,000	18.20	Cornell Dubillier	FAH150000-15-D6
8.625	180,000	34.00	General Electric	86F127
5.625	185,000	34.80	General Electric	86F526
5.875	210,000	19.00	Mallory	CGS214U015X5R
8.625	240,000	29.00	Cornell Dubillier	FAH240000-15-D9

Table 1. This is a list of commercially available capacitors which can be used for C1. All are 3" in diameter and have a rating of 15 V. If you intend to fit everything into the Altair case, then choose a capacitor that is no higher than 5". If you are building a separate supply, one of the larger values may also work.

be safe and conservative.

While this transformer was once available only through SCCS to its members, the improved version can now be bought directly from the manufacturer (see Fig. 1).

The new design differs from an earlier SCCS transformer only in the addition of a third primary tap, the one marked 130 V in Fig. 1. In all other respects the two designs are the same, and for that reason the experiences of those using the old design should be helpful to you. They generally found that all parts of the original early supply as shipped by Mits had to be removed from the computer and scrapped. The only exception is the full wave bridge rectifier, which can be reused on the 16 V lines. Then through careful shopping for parts, especially for a small-sized filter capacitor for the 8 V line (C_1 in the diagrams) and careful layout of parts, it was possible to fit the new supply inside the original case. Careful layout of parts sometimes meant mounting the fan outside the case. I will return to layout questions after discussing the components that make up the supply.

Choice of Parts

Although I've specified one set of parts in the circuit diagrams, it should be clear that a considerable range of values will serve equally well. A discussion of this range will help you keep costs down by utilizing your junk box or surplus stores to greatest advantage.

We'll begin with the bleeder resistors, R1, R2, and R3. If you want them to discharge the capacitors faster, you can reduce their resistance and, in inverse proportion, increase their power rating. In my own supply, I am using 50 Ohms at 5 W for R1, 390 Ohms at 2 W for R2 and R3.

For the part marked BR I suggest you use a full wave bridge rectifier. The one that was used on the 8 V line of the original Mits supply works just fine in my unit, but you can also use others. Get one with a rating of 3 A at 50 V or higher, such as Radio Shack's 276-1171 (4 A at 100 V for $1.49) or Lafayette's 32 P 91036V (10 A at 100 V for $1.59).

For the associated filter capacitors, C2 and C3, I suggest you use electrolytics of 16,000 uF rated at 25 V or higher. This value is arrived at by the rule of thumb that for every Ampere of current drawn, your filter capacitor should have 8,000 uF. This will assure that the ripple component of your output voltage will not exceed 1 V peak-to-peak. And since the transformer is capable of supplying up to 2 A on these two lines, capacitors of 16,000 uF will let you use the full capabilities of the supply. Of course, there's no harm in substituting a higher value, say 20,000 uF, if that is more conveniently available and still fits inside the case. Alternatively, a lower value will probably work, especially if your system doesn't draw the full 2 A on these lines, and to my knowledge no Altair system does. However, a 16,000 to 20,000 uF 25 V capacitor is inexpensive and small enough so that you'll probably find two to fit both the space and your budget.

The 8 V Capacitor

It may be a different story for C1 on the 8 V line, because applying the same rule stated above, this one should be 160,000 uF with a 12 V rating. Since capacitors with a 12 V rating are rare, you will probably have to get one rated at 15 V. Either way, this capacitor is going to be large and you'll have to shop carefully for a usable size (and price). You may find it necessary to make some compromises in order to fit the whole thing into your case, and if so, here are some things to keep in mind:

The value of 160,000 uF was arrived at by the rule mentioned above and the

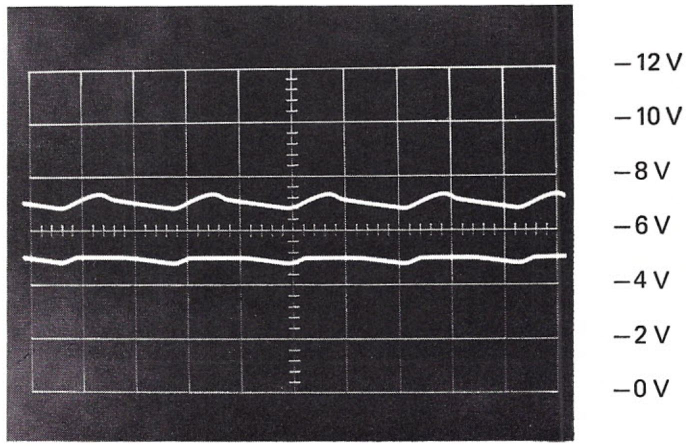


Photo 1. The upper waveform shows the regulator's input going slightly below 7 V at the lowest points. In exactly those places the lower waveform shows the regulator's output to drop out of regulation.

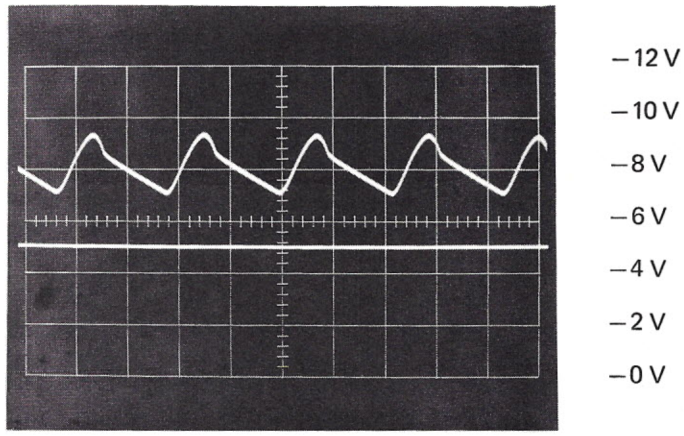


Photo 2. The upper waveform shows considerably more ripple than that of Photo 1, but it never goes below 7 V. That's why the regulator does not drop out of regulation, as is demonstrated by the horizontal line at 5 V, which represents the output voltage.

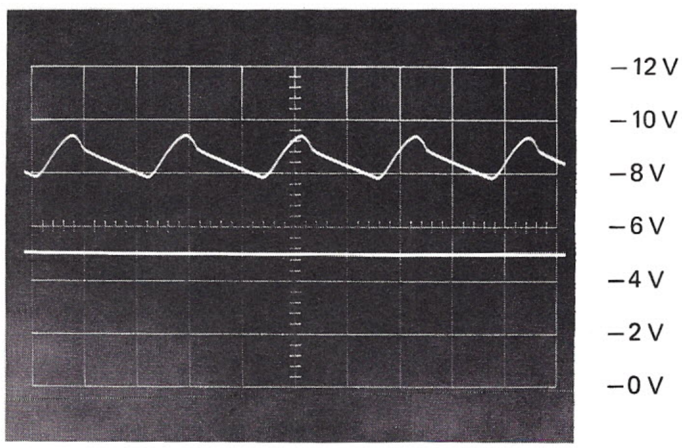


Photo 3. These waveforms show my power supply providing 20 A with C1 of 95,000 uF. Please note the fact that the input never goes below 7.8 V and that the output is perfectly regulated.

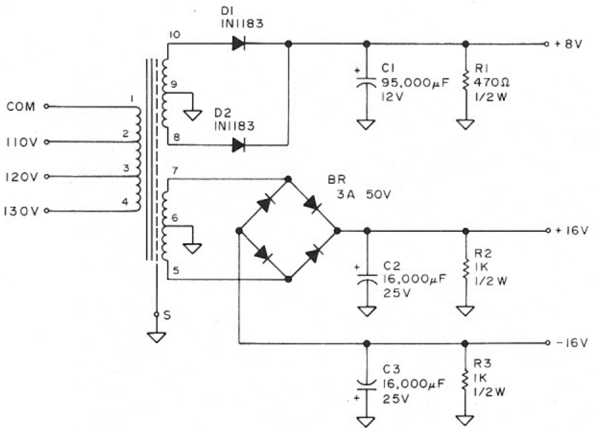


Fig. 1. The transformer is available for $28.00 from the Mijobe Corporation, P.O.B. 775, Claremont CA 91711. A full set of parts is also available for $59.95. This consists of all items in the diagram or their functional equivalent. The diodes provided for D1 and D2 are used at your option. Add 10% for postage and handling, 15% if you are in Canada.

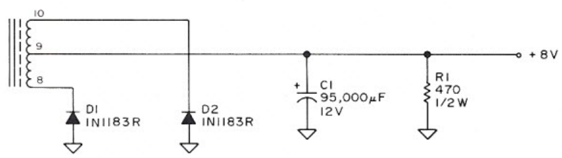


Fig. 2. Layout problems can be reduced by using this circuit with reverse polarity diodes for the 8 V supply. The rest of the circuit remains as in Fig. 1.

observation that the SCCS-designed transformer can supply up to 20 A on this line. This value will keep the ripple component down to 1 V at a current of 20 A. We have to realize, however, that very few people will ever use this full current capability. What if your system draws, say, 12 A on this line? Then you can safely use a capacitor of 95,000 uF and you'll get just about 1 V of ripple. Such a capacitor is readily available for $8.60 and will fit inside the Altair case (see Table 1).

Before going on to the remaining components, let's dwell on the purpose of C1 a bit longer. Its function is to smooth out the waveform of the rectified ac coming from D1 and D2. It has to do this so effectively that the voltage on the 8V line never drops below 7 V, because this is the minimum value that the regulators on your computer's plug-in boards need in order to provide a constant output of 5 V (see Photo 1). In this connection it doesn't matter how large the ripple component is (see Photo 2), it's just crucial that the lowest value of the waveform stay above 7 V at all times.

The waveforms shown in Photo 3 will illustrate an actual system. It shows my supply with C1 at 95,000 uF and a load of 20 A. What I've been calling the ripple component is that wiggly line going between 7.8 and 9.4 V, and that means the value of the ripple component is 1.6 V peak-to-peak. If I had been using 160,000 uF for C1 instead, this waveform would have become shallower and the ripple component only about 1 V peak-to-peak. Notice, however, that even with C1 at 95,000 uF the actual voltage on the 8V line never drops below 7.8 V, and therefore the needs of my computer are adequately met. There's even .8 V of protection or headroom in this design so that if my house voltage drops momentarily, as when my refrigerator or air conditioner starts, the 8 V line is still not likely to drop below 7 V.

The point of all this is to assure you that even with a capacitor of 95,000 uF — as specified in the diagrams — you'll have a fully usable supply. If you can fit in more capacitance than this, so much the better, and if you're not going to draw the full 20 A, you can even use less. I will have some more comments on the selection of C1 below.

The Rectifier Diodes

As for the remaining rectifier diodes, D1 and D2, I suggest that you make them stud-mounting types with a voltage rating of 25 V or higher. Make the current rating at least 25, preferably 35 or even 40 A, especially if you're planning to use the full current rating of this winding. Types 1N1183 (35 A at 50 V) and 1N1183A (40 A at 50 V) are excellent choices.

The circuit diagram of Fig. 1 will require that you mount these diodes with insulators. If you buy diodes with reverse polarity, i.e., with the negative terminals connected to the stud (such as the 1N1183R or 1N1183RA), then you don't need insulators for the circuit in Fig. 2, which shows the 8 V winding only.

Either way, these diodes will have to be mounted firmly — preferably with silicone grease — on a suitable heat sink. I suggest you mount them on the aluminum bracket that held the original Mits power supply board or, if you are using the alternate diagram of Fig. 2, you can mount them on the back panel after scraping off the paint. If stud-mounted rectifiers are not conveniently available, you can substitute a 25 A full wave bridge, such as those advertised by some vendors in *Kilobaud*. In this case, you will connect only half of the bridge according to either circuit diagram, depending on which half you decide to use.

Look Out for Ground Loops

Whenever you build a power supply providing more than a couple of Amperes of current, ground loops can be a real problem unless you take proper precautions. In fact, ground loops are amoung the most frequently misdiagnosed causes of transient malfunctions in electronic equipment. The following precautions are related specifically to the 8 V line, since it can supply rather high currents, but the same principles can be applied to the other lines as well. They are simply good construction techniques.

The idea is to prevent high currents from causing voltage differences between various ground points in a piece of equipment. This undesirable condition is usually brought about by using the chassis as the ground lead not only for various signals but also for the power supply. And then, if the power supply provides a good deal of current, a voltage drop will actually

Capacitance in uF	Price in $	Manufacturer	Model number
16,000	5.45	Cornell Dubillier	FAH16000-25-E3
19,000	8.70	General Electric	86F141
20,000	5.60	Mallory	CGS203U025V3C
20,000	5.20	Mallory	CG203U025V4C
22,000	9.50	General Electric	86F540
24,000	6.20	Cornell Dubillier	FAH24000-25-B3

Table 2. These capacitors are good candidates for C2 and C3. They are all rated at 25 V and are less than 5" high, so that dimension does not need to be given. The first unit on the list is 1.75" in diameter, all others are 2". The prices for equivalent capacitors at surplus stores can easily be half of those listed in the tables.

occur over the path of that current in the chassis, and this voltage will then be superimposed on the signal levels grounded along this path. I hope that's clear.

You have two weapons against this problem: (1) avoid using the chassis as a high-current ground return; (2) reduce the impedance of your ground return. This is a simplification of what's involved, but it's close enough to get us started.

If you use the diagram of Fig. 1, then be sure to connect the negative terminal of C1 directly to terminal 9 of the transformer with heavy gauge wire (no. 14 or larger). Connect the ground lead going to the Altair bus to this same wire at C1. Finally, connect a lead anchored to the chassis to the same place. This is an application of weapon 1. Be sure to use heavy gauge wire (as above) for all current-carrying leads in the 8 V supply, including the positive leads.

If you use the alternate diagram of Fig. 2, then do *not* depend on chassis grounding alone to carry all that current. Instead, let's use weapon 2 and reduce the impedance of our ground return. Bolt heavy gauge wires (as above) under each diode; then twist and solder these two leads together (see Photo 4). Run the continuation of one of these wires to the negative terminal of C1 and the other directly to the ground line of the Altair bus.

Optimal Layout

While many different layouts will let you get all parts inside the Altair case, thermal considerations will suggest that you put the hottest components closest to the fan. In fact, if you decide to mount the fan outside the case, you can achieve a thermally optimal design and even avoid blocking any of the connector holes on the back panel. Photos 4 – 7 show the construction sequence for my own supply.

Photo 4 shows the layout of parts on the back panel, using reverse polarity diodes and the circuit of Fig. 2. Please note that the transformer mounts directly over the fan hole with countersunk stove bolts. The diodes are mounted with silicone grease directly to the back panel after the surrounding paint is removed. The heavy ground wires mentioned above are clamped beneath them. The full wave bridge rectifier from the original Mits supply is mounted to the right, again with silicone grease and after removing the surrounding paint. The fuse holder, power line cord and switch are mounted directly below. The hole just below the lower-right leg of the transformer is where the wires going to the fan will be routed. Photo 5 shows most of the wires installed.

Photo 6 shows the relationship of the back panel to the aluminum bracket holding the capacitors, and Photo 7 shows the completed installation. Please note that there are four capacitors. One is 3″ in diameter, another 2″ and two are 1.75″. The first two are connected in parallel to form C1. Within this same space you can also fit capacitors with the following diameter combinations: one 3″ and two 2″; two 2″ and two 1.75″; four 2″. The reason I list these is to guide your shopping for combinations that will fit. The length of these capacitors should not exceed 5″ as measured on the side of the can. That will leave 5/8″ clearance for the wiring.

To make your shopping a bit easier, Tables 1 and 2 show the parts numbers for several likely candidates for the capacitors. The data are taken from three manufacturers' catalogs and are current as of April 1977.

That takes care of my specific construction comments. I assume that you will know about proper fusing procedures and safety rules, that you'll observe all necessary polarities of capa-

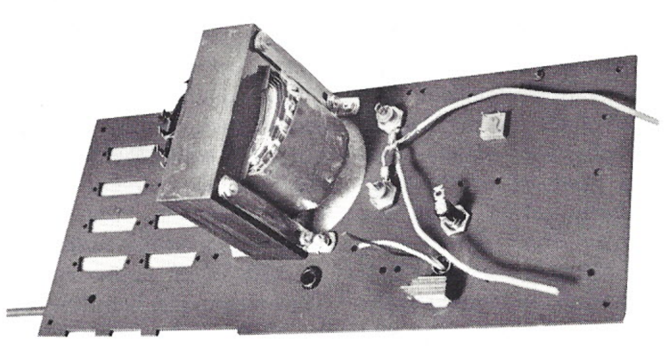


Photo 4. Detail showing parts placement on the rear panel of the Altair 8800. The transformer mounts with countersunk stove bolts directly over the fan cutout. The stud-mounted diodes must be reverse polarity types and the diagram of Fig. 2 must be used with this layout.

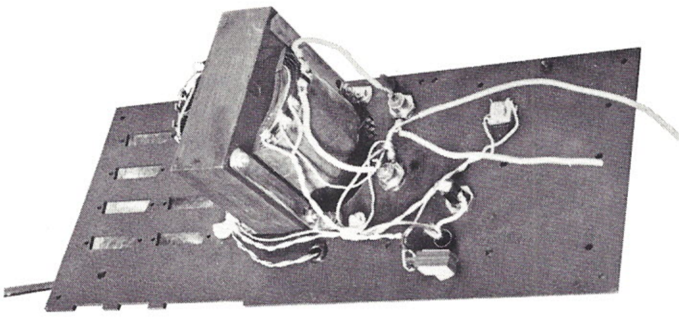


Photo 5. Note the routing of wires. Those coming through the hole below the lower-right leg of the transformer are connected to the fan.

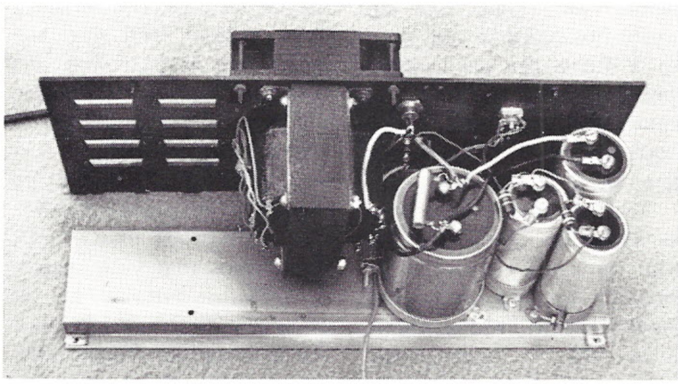


Photo 6. The fan is mounted outside the case in order to fit all other parts comfortably inside. That way the connector cutouts on the left are not obstructed.

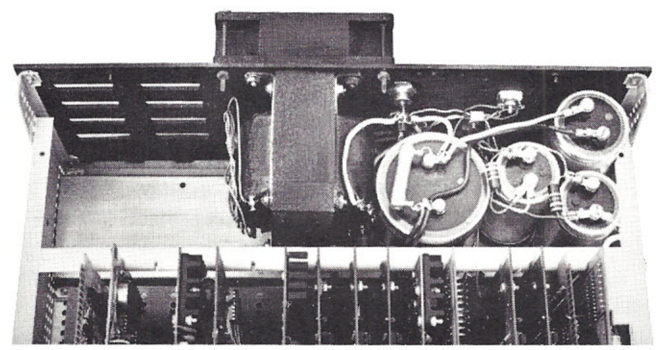


Photo 7. The completed power supply as installed in my Altair. C1 consists of the two larger capacitors connected in parallel. Please note the close placement to the fan of the principal heat-generating components, namely the transformer and the stud-mounted diodes. Everything runs nice and cool with this layout.

citors and rectifiers, that you'll test the outputs before connecting them to the bus, that you'll connect the output lines to the correct bus lines and, finally, that you'll read all of these instructions before doing any of this. Don't forget that your Altair has separate 8 V and ground leads going to the front panel which must also be connected to your new 8 V supply line and bus ground.

Which Tap Do I Use?

Now you're almost ready to make the whole thing work, but first I have to tell you *something important about the 130 V tap of the new design.* Its purpose is to prevent an over-voltage problem in systems drawing modest amounts of current. Users of the old design reported that the actual voltage on the 8 V line could go as high as 12 V when they had only two boards in their computer. While the three-terminal 5 V regulators can handle this, they did get pretty hot. Besides that, both 16 V lines also went correspondingly higher, and since some boards use zener diode regulators on these lines, real care had to be exercised.

To overcome this problem we added the 130 V tap. This will be used in small to medium systems only, and the 8 V line will not go higher than 10 V even without a load. But here is an important point to keep in mind: *That section of the primary that is brought out as the 130 V tap (terminal 4) is wound with a smaller gauge wire than the rest of the primary.* This was done because there was only a small amount of space still available for additional windings on the old design, and because we did not want to increase the size so it would still fit inside the Altair. Besides, we wanted to keep costs down to a reasonable level; therefore this design compromise was adopted.

This means that you must not attempt to get full power out of the transformer while using this tap or overheating and failure may result. Your most practical protection against such damage is just to put a *2 A slow-blow fuse* into the primary circuit. When using this tap, the transformer can readily supply up to 120 W of power safely. That means you can draw up to 12 A on the 8 V line and up to a total of 1.5 A on the other two.

In practice this will work out just fine. By the time you're drawing 12 A the output voltage will be low enough so that you'll want to change to the 120 V tap anyway, and from there you can draw full power. The only thing you have to keep in mind is that during your initial setup procedure you don't try to run a large system on the 130 V tap. Clear?

Now let's get back to making the whole thing work. We have to select the primary tap which is correct both for the current requirements of your system and for the actual line voltage. I suggest that you begin by connecting the input power to the 120 V primary tap and using a 4 A slow-blow fuse. Now measure the actual voltage on the 8 V line. If it is less than 7.8 V, that means that your system is drawing quite a bit of current and/or the line voltage is quite low. Measure the line voltage, and if it is less than 115 V, then reconnect to the 110 V tap. You should use this tap only if the line voltage is consistently low in your area. If, on the other hand, the 8 V line measures between 7.8 and 9.2 V, then the 120 V tap is the correct one to use. Finally, if the measured voltage is over 9.2 V, then reconnect to the 130 V primary tap and *switch to a 2 A slow-blow fuse.* This will give your transformer the protection it needs. Should the fuse ever blow, you know either that something is shorted or that you have to use the 120 V tap and a 4 A fuse.

Through the procedure described in the preceding paragraph you'll be able to adjust the 8 V line to the requirements of your system. Ideally, you want the measured voltage to be close to 7.5 V and always to stay there. On a scope trace, you want the lowest part of the waveform to stay just above 7 V. But since your line voltage may sag at times, it is safer to try for a bit higher a voltage, say between 8 and 9.5 V. That will increase your headroom and still not cause excessive heat dissipation. If you ever change your system extensively by adding, subtracting or substituting boards, you have to repeat the procedure above.

Protect Your Zener Diodes!

At this point you've taken care of the 8 V line. But what about the other two? In most cases they will have taken care of themselves, but it is best to double check, so do this: If the measured voltage on these lines is less than 19 V, then you should have nothing to worry about. If it is higher, you may do the following: Determine if any of your boards use zener diode regulators on the +16 or -16 V lines. IF not, then again you should have nothing to worry about, because three-terminal regulators will have no trouble with a modest over-voltage on these lines.

If, on the other hand, you do have zener regulators (e.g., on the original Altair CPU board or on the Tarbell cassette interface board and some others), the additional power dissipation caused by an over-voltage condition may eventually cause problems, and I suggest precautionary measures.

The cheapest such measure is to add a resistor on each board involved in series with the power line involved and removing no existing parts. Choose a value that will drop the voltage down to 16 V.

Another approach is to add rectifier diodes (2 A minimum rating) in series with the supply line before it is connected to the bus. Each diode will cause a drop of about .7 V. Use as many diodes as are needed to get close to 16 V.

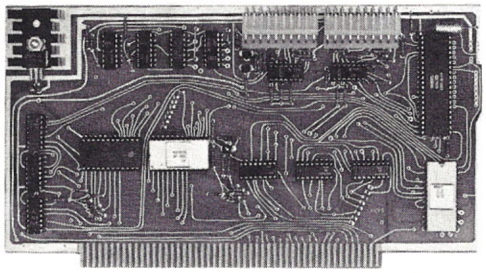
A more elegant solution is to replace the entire zener diode regulator itself with a three-terminal regulator. This is the solution I recommend, but the others will also work well after you've figured out the proper resistance and power rating, or the correct number of diodes. In either of these cases, you may have to make new adjustments if you make extensive changes in the power requirements of your system. With a three-terminal regulator, on the other hand, you won't have to worry about it any more.

A Final Caution

The capacitors in your new supply are considerably larger than those of the original supply. That means they will discharge more slowly after the power is turned off, and that in turn means you have to wait longer before removing or inserting boards into the bus. Otherwise damage will result. Believe me, it's true! Determine safe timing with a voltmeter for each line, and then always observe the longest of them. One user reports that he waits for the fan blades to stop turning. That protects both his boards and his fingers.

There you have it, a rather detailed explanation of how to use your new SCCS-designed transformer. Just about all of my suggestions are intentionally conservative. They assure that you won't get into any trouble and that you'll have a reliable long-lasting supply. In the name of all those who will use, or are already using, this transformer, I want to thank Chris Marshall for a good idea and Michael Laub of Mijobe Corporation for a well-tuned design. Thanks also to Gerhard Clausing for the photos. ■

THE LATEST IN TAPE SYSTEMS



MODEL CC-8 – $175.00 4800 BAUD CASSETTE RECORDER

An ASYNCHRONOUS NRZ type recorder with remote motor start/stop. Error rate 10^8 at 4800 BAUD. Can be used from 110 to 4800 BAUD into a UART or "Bit Banger PIA" – no clocking required. This is not an audio recorder. It takes RS232 or TTL signals from the terminal or computer and gives back the same signals. No audio interface is used. Motor start/stop is manual or through TTL or RS232 signals.

Tape speed is 3.2"/second nominal; 1.6"/sec. optional. 110 volt, 60 Hz, 5 Watts. (220 Volts on special order). Can use high quality audio cassettes (Philips type) or certified data cassettes. Can be used in remote locations from a 12 Volt battery.

Recommended for DATA LOGGING, WORD PROCESSING, COMPUTER PROGRAM RELOADING and DATA STORAGE. Especially recommended for 6800 systems, 6502 systems, 1800 systems and beginners with the 8080 systems. Manual control except for motor start/stop. 6800 or 8080 software for file or record searching available on request with order. Used by major computer manufacturers, Bell Telephone and U.S. Government for program reloading and field servicing. AVAILABILITY – Off the shelf.

2SIO(R) CONTROLLER – $190.00 ($160.00 Kit) PROVIDES MONITOR AND TAPE SOFTWARE IN ROM TERMINAL AND TAPE PORTS ON SAME BOARD CONTROLS ONE OR TWO TAPE UNITS (CC-8 or 3M3A)

This is a complete 8080, 8085, or Z80 system controller. It provides the terminal I/O (RS232, 20 mA, or TTL) and the data cartridge I/O, plus the motor controlling parallel I/O latches. Two kilobytes of on board ROM provide turn on and go control of your Altair or Imsai. NO MORE BOOTSTRAPPING. Loads and Dumps memory in hex on the terminal, formats tape cartridge files, has word processing and paper tape routines. Best of all, it has the search routines to locate files and records by means of six, five,and four letter strings. Just type in the file name and the recorder and software do the rest. Can be used in the BiSync (IBM), BiPhase (Phase encoded) or NRZ modes with suitable recorders and interfaces.

This is Revision 7 of this controller. This version features 2708 type EPROM's so that you can write your own software or relocate it as desired. One 2708 preprogrammed is supplied with the board. A socket is available for the second ROM allowing up to a full 2k of monitor programs.

Fits all S100 bus computers using 8080 or Z80 MPU's. Requires 2 MHz clock from bus. Cannot be used with audio cassettes without an interface. Cassette or cartridge inputs are RS232 level. AVAILABILITY – Off the shelf.

MODEL 3M3A – $220.00 50 KILOBAUD CARTRIDGE RECORDER

This is a self-clocking (1/1) high speed recorder. Loads BASIC in under 2.0 seconds. Recording is done at 19.2 Kilobaud. Playback at 50 Kilobaud. File or record searching is done at 50 Kilobaud and loading is automatic. Worst case access time about 2 minutes for up to 2 megabytes on the 3M Data Cartridge.

Tape speed 10"/sec. on record, up to 30"/sec. on playback. Records one clock track and one data track on each pass (2 passes). Recording is NRZ unencoded from RS232 or TTL signals.

This recorder requires one parallel port for motor control, and one serial port for data and clock. (Cannot be used with UART's or UART boards such as the 3P+S). Used with USART's, ACIA's or other 1/1 clocking I/O devices under software control only. No manual controls. Software for 8080 and 6800 available. Power supply is built in, 110 V, 60 Hz. 220 V, 50 Hz for export.

OVERSEAS: Export Version 220 volt 50 hz. Write factory or: Megatron-Datameg, 8011 Putzbrunn, Munchen, Germany; Nippon Automation 5-16-7 Shiba, Minato-Ku, Tokyo, Japan; Hobbydata, FACK 20012, Malmo, Sweden; G. Ashbee, 172 Ifield Road, London SW 10-9ag; Trintonics, Ltd., 186 Queen Street W., Toronto, Ontario, Canada; EBASA, Enrique Barges 17, Barcelona 14, Spain; ARIES, 7, rue Saint Phillipe du Roule, 75008 Paris; Microlem 20131, Milano, Italy; Eagle Electric, Capetown, S. Africa.

Canadian Distributor:
Trintronics Limited
186 Queen Street West
Toronto, Canada M5V 1Z1
Tel: (416) 598-0262

NATIONAL MULTIPLEX CORPORATION

BANKAMERICARD master charge N-1

3474 Rand Avenue
Box 288
South Plainfield NJ 07080
Phone: (201) 561-3600
TWX: 710-997-9530

For U.P.S. delivery, add $3.00. Overseas and air shipments charges collect. N.J. Residents add 5% Sales Tax. WRITE or CALL for further information. Phone Orders on Master Charge and BankAmericard accepted.

Robert M. Tripp, Ph. D.
The Computerist
P.O. Box 3
S. Chelmsford MA 01824

Is the KIM-1 For Every-1?

... find out if it's for you!

MOS Technology was recently purchased by Commodore Business Machines, Inc., 901 California Ave., Palo Alto CA 94304. Commodore will be manufacturing and distributing the KIM product line, and is expanding the production facilities to double or triple the number of KIMs produced. — Ed.

Of course not! No single microcomputer can serve everyone's requirements. But, the MOS Technology KIM-1 microcomputer is a well integrated package that has features which have appeal to hobbyists, educators, and industrial users. How much do you think it will cost you to buy this complete computer system with the following features:

- 6502 MOS Technology Microprocessor.
- HEX Keypad plus seven Control Keys.
- six digit LED Display.
- 110 to 2400 baud 20 mA Current Loop Teletype Interface.
- 800 baud Audio Cassette Interface.
- 2K ROM Monitor which works with the Keypad/Display or Terminal.
- over 1K bytes of RAM.
- two independent Programmable Interval Timers.
- thirty (30) Programmable I/O Lines.
- extensive Hardware, Programming, and User Manuals.
- capable of expanding the I/O Ports.
- capable of expanding the Memory to a full 65K bytes.
- completely assembled and tested.

How much does it cost? Five hundred dollars? Eight hundred? More? Less! Would you believe $245? Look at the features again. That's quite a bundle of goodies for the price.

If you have priced other systems with comparable features, you are probably wondering what the catch is. "This is a new, small, fly-by-night operation which will either have gone out of business or raised its prices by the time I place my order. Right?" Wrong! The history of the KIM-1 is a bit unusual. MOS Technology which manufactures the KIM-1 also manufactures the 6502 microprocessor and other related microcomputer oriented chips. When their 6502 was first ready to be introduced to industry, they decided to make a powerful "evaluation kit" which, unlike those offered by most other vendors would be completely assembled, tested, and would be capable of performing real applications. There are now over seven thousand KIM-1s in the field. These are being used mostly by industry, but many units are also being used for educational purposes and by computer hobbyists.

The hobbyist appreciation of the KIM-1 has been a little slow to develop for two main reasons. First, there has not been very much published about the KIM-1 in the national computer hobbyist magazines (which is part of my motivation for writing this article). Second, until recently, the dealer discount structure was such that very few dealers were interested in handling the KIM-1. This has been changed and a lot of computer stores are starting to carry the KIM product line. A number of computer clubs now have formed KIM-1 sub-groups, and there is a national publication, *KIM-1/6502 User Notes*, which is hobbyist oriented and has a rapidly growing subscription list — currently over eight hundred. Assuming that about 25% of the KIM-1s sold to date have been to hobbyists, then there are about two thousand currently in hobbyists' hands, and perhaps one hundred or more being added each month. This is a significant portion of the computer hobbyist population. I do not know how extensive the international distribution of KIM-1s is, but I have received orders for software from Germany, Italy, Sweden, Taiwan and Kuala Lumpur, Malaysia!

Since you have read this far, you are probably at least considering the KIM-1 for your own. So let me discuss the features in detail. The 6502 has a good general purpose instruction set. in many ways similar to the 6800. It has one of the best sets of addressing modes available. These include Relative Branching, Indexed Indirect and Indirect Indexed modes useful in table processing, Stack Addressing, and others. The 6502 microprocessor has been selected by a number of independent companies for use in their hobbyist oriented systems. These include the APPLE-1 by Apple Computer Company; BABY! by STM Systems; the Challenger by Ohio Scientific Instruments, and Micromind by ECD Corp. to name a few. These are all assembled systems. The 6502 is also found in a number of kit systems.

The keypad has twenty-three keys and a slide switch. The keys include the sixteen hex digits and seven program dependent functions. Two of the keys are tied into the interrupt structure providing "maskable" and "non-maskable" interrupts to be generated from the keypad. The keypad, in conjunction with the LED display and the ROM monitor, make it possible to enter programs directly into memory, to execute programs, and to do extensive program debugging including single-step testing. All this without an expensive front panel or external terminal.

The LED display consists of six independent seven-segment LEDs. These are normally used to display hex data: four digits of address and two of memory contents. These same LEDs may be used to output alphabetic messages, chess board coordinates, decimal calculator values, and so forth.

If you are lucky or rich enough to own a Teletype compatible terminal you can connect this directly to the KIM-1. The KIM-1 hardware provides a 20 mA current loop interface. The KIM-1 monitor provides the software to drive the terminal at rates from 110 to at least 2400 baud, with some users reporting good transmission at 4800 baud and reasonable transmission with occasional glitches at 9600 baud. Baud rate is automatically determined by the software. There are no jumpers to move or switches to set.

In addition to providing the standard commands (enter, modify, execute, etcetera debugging) the monitor also supports punching and reading paper tape. The user simply sets the starting and ending addresses for the dump and the monitor takes care of formatting the data, calculating check digits, and transmitting the data to the terminal. This support makes it easy to save and load programs via paper tape. Your terminal may be a hardcopy or video type. KIM-1 doesn't care.

The "piece de resistance" of the KIM-1 is its built-in audio cassette interface. An audio cassette is the type of recorder you use to record and listen to music. Nothing simply storing and retrieving programs from standard audio cassettes is a great benefit to the average hobbyist.

The 2K ROM monitor, which is contained in the ROM portion of two 6530 multi-purpose chips, is an integral part of the KIM-1 system and has a number of

How much does it cost? Five hundred dollars? More? Less! Would you believe $245?

special. The recording technique implemented in the KIM-1, and described in some detail in the KIM-1 User Manual, is very conservative and provides tapes that may be readily interchanged between all KIM-1s, and most types, brands, and qualities of cassette recorders. (Tapes are not interchangeable with any other recording system). I have distributed over three hundred tapes recorded directly from my KIM-1 with only a few problems. These problems have all turned out to be due to out-of-alignment cassette recorders.

While the tape dump routine of the KIM-1 monitor puts data out at the tediously slow rate of about three minutes per 1K of memory, there is a software routine available called Supertape which will dump KIM-1 compatible tapes at six times the standard rate or about thirty seconds for 1K bytes. These tapes may be loaded via the KIM-1 monitor tape load routine at the higher rate with no modifications. Other tape routines are possible (and are documented in HELP) which work with the KIM-1 hardware and produce data transfer rates at 800 baud or 100 bytes per second. The capability of clever and useful functions. It provides the standard capabilities of examining and modifying memory locations from either the keypad/display or terminal. It also supports single-step program execution for debugging purposes. Whenever a program is stopped, either via the stop (ST) interrupt key or while in single-step mode, any memory location can be examined and modified. To resume processing there is a program counter (PC) key which restores the value of the program counter before restarting the program with the execution (GO) key. The 6502 has a BREAK instruction which generates a software controlled interrupt. This may be used in conjunction with the monitor to insert a trap into a program for debugging. The monitor also contains all the software required to control the keypad/display, terminal, and audio cassette. Many of the monitor's routines may be used by user generated programs, especially to perform standard input/output functions. The ROM even has a special program for fine tuning the audio cassette interface, should the need ever arise.

There are two sections of read/write memory on the KIM-1. The main RAM is 1K (1024) bytes of 2102 type static RAM. In addition, the 6530 multipurpose chips each contain 64 bytes of RAM, for an additional 128 bytes total. Of these extra memory bytes, 25 are normally reserved for use by the monitor and 103 bytes are always available to the user. While a total of 1152 bytes of RAM may not seem like much memory, you can actually do quite a bit with it. I will list a few programs which operate in this amount of memory in the software section.

If you require more memory for your application, it is simple to add memory to the KIM-1. MOS Technology offers two completely assembled and tested memory boards for direct connection to the KIM-1 with no additional buffers. The KIM-2 is a 4K RAM and the KIM-3 is an 8K RAM. These boards both use the 2102 type static RAM chips. One of these boards may be interfaced to the KIM-1. If you require more than 9K bytes of RAM, MOS Technology offers the KIM-4 which is a board with buffers and connectors that permit the addition of memory up to a total of 65K bytes for the system. The additional memory may be any combination of RAM and ROM. Some of the ROMs to be offered by MOS Technology include a floating point math package and an editor/assembler package.

Each of the 6530 multipurpose chips includes a programmable interval timer, which may be set from a few microseconds to a quarter of a second. They may be tested under program control or may be set to cause an interrupt on completion of the specified time interval. These two timers take a tremendous burden off of the software for many real-time programs, and can be very useful in programming clocks, music generators, and the like.

Communication with the

"outside world" is handled by the peripheral interface ports of the 6530 multi-purpose chips. Each chip handles 15 input/output lines. One set of I/O lines is used by the KIM-1 to control the keypad, display, terminal interface and audio cassette interface. The other set is available to the user. These are configured and programmed as standard parallel interface adapters (PIA). They may be used to turn devices on and off, to sample external devices, and so forth.

The documentation which comes with the KIM-1 is pretty good. The *KIM-1 User Manual* includes the information necessary to attach your audio cassette and terminal; descriptions and examples of using the monitor in both the keypad/display and terminal modes; a simple programming example; a "real application" example which includes using the programmable I/O ports; info on expanding your memory and I/O capacity; and the complete monitor source listing. The *Programming Manual* is a 170+ page document which covers the 6502 instruction set, addressing modes, peripheral programming, and other pertinent materials. The *Hardware Manual* contains over 150 pages on the 6502 Microprocessor, the 6530 Peripheral Interface/Memory Device, and the 6520 PIA (which is not used on the KIM-1). You also get a multi-colored wall chart, programmers card, etc.

That pretty much covers the KIM-1 system. You must supply the power, +5 volts at about 1.2 Amps and -12 volts at about 100 milliamps (the -12 being required only if you are using the audio cassette and may be supplied by a battery). You can build your own power supply following the circuit diagram provided in the *KIM-1 User Manual*, or, *The Computerist* has a new power supply designed specifically for the KIM which can power the KIM-1 and additional memory. It costs $40 for the completely encased unit. Or a surplus power supply (adequate for the minimal KIM-1 but no additional memory) is available for $25.

Now you have your KIM-1 and it's powered up. What would you like to do?

Now you have your KIM-1 and it's powered up. What would you like to do? Play games? *The Computerist* offers two games packages, each of which comes with the programs on a Supertape cassette tape and includes complete documentation and source listings. "PLEASE" is an assortment of games and demonstrations, including a 24-hour clock, a millisecond timer, the Shooting Stars puzzle, the Mastermind game, Hi-Lo game, a simple adding machine, an intoxication tester, and more. It runs on a minimal KIM-1 system and costs $10. The second package is MicroChess which plays a pretty good game of chess on the minimal KIM-1. It was written by Peter Jennings and is available for $15.

When you are done playing games and are ready to put your KIM-1 to work, you can get "HELP," a series of application packages which work with the minimal KIM-1, a terminal, and a pair of audio cassette recorders with relays for turning them on and off under program control. The HELP packages include a source and text editor, a mailing list preparation/printing package, a form letter generator/printer, and an information retrieval package. Each package comes on a Supertape cassette tape and includes complete documentation and source listing. HELP is written in a high level language which permits the user to write his own applications and/or customize existing applications to suit his particular requirements. They cost $15 per package, and a relay package containing all of the components (less mounting board) to control two cassette recorders is available for $10, all from *The Computerist*.

Add 4K bytes of RAM and you can run Tom Pittman's Tiny BASIC. He has a version specifically for the KIM-1. There are a number of groups that are actively developing software for the 6502 and the KIM-1. Lack of software has somewhat limited the growth of the KIM-1 as a hobby computer, but availability is rapidly improving.

One other factor that has limited KIM-1 growth has been the fact that it does not conform to the Altair bus structure. Since there are a lot of very nice peripherals which are Altair compatible, similar capability for the KIM-1 would be valuable. Forethought Products has just announced the KIMSI S-100 Interface/Motherboard which connects to any unmodified KIM-1 computer and converts its signals to the Altair bus format. The board also contains 8-100 pin slots making it a useful motherboard as well. The price is $125 in kit form and $150 assembled. The use of this board will permit the simple addition of a wide variety of peripherals to the KIM-1 and greatly extend its usefulness to the hobbyist.

Are you hooked? Since computer stores are now carrying the KIM-1, you can probably see one in action locally. Or some other computerist in your area probably owns one and would be happy to show it off. Have fun. ■

1. MOS Technology, 950 Rittenhouse Road, Norristown PA 19401, 215/666-7950, Manufacturer of the KIM-1, KIM-2, KIM-3, . . ., 6502, 6530 . . .

2. *KIM-1/6502 User Notes*, c/o Eric C. Reknke, 425 Meadow Lane, Seven Hills OH 44131. Independent hobbyist magazine covering the KIM-1 and 6502. Published every 5 to 8 weeks. It contains software routines, games, notes, announcements, etc. ($5 for issues 1-6, $8 foreign subscriptions).

3. The Computer Shop, 288 Norfolk St., Cambridge MA 02139. 617/661-2670. 4K RAM kit which can be used with the KIM-1. $74.50 with 2102 type static RAM.

4. The Computerist, P.O. Box 3, S. Chelmsford MA 01824. 617/256-3649. Creator and distributor of the PLEASE and HELP software packages, MicroChess, and a KIM-1 power supply and surplus power supply. *The Computerist* is a monthly publication dealing with microcomputers in the New England region from a hobbyist point of view ($6/year).

5. Forethought Products, P.O. Box 386-A, Coburg OR 97401. Manufacturer of the KIMSI S-100 Interface/Motherboard.

6. Newman Computer Exchange, 1250 N. Main, Ann Arbor MI 48104. Distributor for a composite video peripheral for the KIM-1 ($239).

7. The 6502 Program Exchange, 2920 Moana Lane, Reno NV 89509. Games and Utility software for 6502 based systems.

8. Johnson Computer, P.O. Box 523, Median OH 44256. KIM-1 related hardware and software.

9. Tom Pittman, P.O. Box 23189, San Jose CA 95153. Tiny BASIC which will run in 2K bytes on a KIM-1 with additional memory ($5).

Table 1. References and where to find it.

COMPUTERMANIA T.M.

THE BIGGEST EXPO YET! AUG 25 26 27

THU. 1-10PM FRI. 1-10PM SAT. 9AM-9PM

Did you miss the Faire in San Francisco? Don't let that happen again! Plan to come to COMPUTERMANIA in Boston and see everything there is to see.

Have you been wondering about the new Heath microcomputer system? The plans are to debut this at COMPUTERMANIA. Maybe you've been wanting to see what Radio Shack is coming up with? They're aiming at the COMPUTERMANIA for the first showing of their new computer. In fact, just about every firm in the business is working hard this summer to have something to get your attention at COMPUTERMANIA.

VISIT BOSTON

Unlike other cities, Boston is worth a visit in its own right . . . the Pier is just between two of the most famous Boston restaurants and across the street from two others . . . Anthony's Pier Four, Jimmy's Harborside, the No Name Restaurant and the old Union Oyster House. We'll have a lot of information for you on restaurants you won't want to miss.

Boston is historic too, and fantastic to visit for that reason alone. This is where it all got started.

You'll hear talks by the top people in the field and get to ask questions about the new equipment. The surplus people will be there with all sorts of great bargains. See the latest in floppies, in cassette systems, and get some real deals on tapes and disks.

BOSTON COMMONWEALTH PIER

Air Conditioned

GOOD FOOD TOO

In addition to the nearby famous restaurants (some very reasonable in price) there will be outstanding food available at COMPUTERMANIA . . . arrangements have been made for Mexican food, for Chinese food, etc . . . in addition to the usual pizza and hot dogs. You'll have fun.

If you are driving, there is room for thousands of cars right across from the Pier.

WHAT SYSTEM TO BUY?

Before you invest $1000 or more in a home computer system (or a small business system) you surely want to take a very close look at everything that is available. It is very difficult to tell what systems can do just by reading the ads or the literature . . . you really need to see them and sit down and give them a try . . . and this is what COMPUTERMANIA is all about. The newest in hardware will be there, all set up with programs you can check out and try.

If you're into games try out the Star Trek and see how good it is. If you want to go into business printing out statements for local businesses, see what kind of a job the systems will do with that. The people who have designed and built the systems will be there so you can ask them questions . . . and many of them will be putting on illustrated talks about their systems.

How good are some of the new printers? You can only tell if you see them at work. How fast (or slow) are the systems? You can tell a lot more about that by trying them than reading about them. How easy are some of the new keyboards to use? How about the color graphics you've been reading about? You certainly want to see what is being done with these. And where else can you hear the many music generators and music I/O systems that are coming out?

SAVE TIME AND MONEY

It will all be at COMPUTERMANIA . . . so send in your pre-registration now and save $2 over the cost at the gate.

You can send cash, check, money order, stamps, or charge your tickets to a Master Charge, BankAmericard or Amex card. Further, you can grab a phone and call in your ticket order on our WATS line: 800-258-5473 during business hours.

SEE CALCULATORS THRU MINICOMPUTERS

In addition to hobby computers you'll see the latest in calculators right on up through the low end minicomputer small business systems. See it all at COMPUTERMANIA!

OVER 300 EXHIBITS AND 25,000 PEOPLE EXPECTED!

DEMONSTRATIONS*FORUMS*TALKS*OVER 250 EXHIBITS*PRIZES

TICKETS AT DOOR: $12.00

POLL • POLL • POLL • POLL • POLL • POLL

MY INTEREST IS IN:
☐ Calculators
☐ Microcomputers
☐ Business Uses
☐ Video Games
☐ Hobby Computing
☐ Industrial Control

Occupation ____________

Equipment now in use (make & model) ____________

Thinking of buying ____________

Present investment $ ____________

Expect to spend during next year? $ ____________

PRE-REGISTRATION PRICE – $10 UNTIL AUGUST 1st

☐ YES! Send me ____ admission tickets at $10 each.
☐ Check enclosed ☐ Credit card below
☐ BA ☐ AMEX ☐ MC – Interbank # ____________
Card # ____________
Expir. date ______ Signature ____________
Name ____________
Address ____________
City ____________ *State* ______ *Zip* ______

Computermania Tickets
Peterborough NH 03458
or call Toll Free (800) 258-5473

KB 8/77

Jim Huffman
Hufco, Inc.
P.O. Box 357
Provo UT 84601

Electronic Design by Computer

... simplify your next project!

We've been wanting applications software, right? Well, wait until you see this one. It's a beaut! Jim has put together a collection of computer-aid design (CAD) routines for the electronic experimenter like you've never seen! Most of the routines were developed for the active do-it-yourself ham, or perhaps even an analog design engineer.

Two points Jim made in his article bear elaboration: First, he feels he has simply laid the groundwork with the article. This package is without end. *We'll probably see routines to be added to the "Hufco CAD Package" in the pages of* Kilobaud *for many moons to come. I would think some digital design programs would be appropriate for openers.*

Second, Jim will be the first to admit that he isn't the World's Greatest Programmer. I sure wouldn't want to hear any voices which are too critical of his work, though. After all, he did sit down and do it! — John.

The following is a description of a series of utility programs which fall into the category of computer-aided design programs. They are implemented in BASIC, both to exploit the mathematical capabilities of BASIC and to provide universal application of the programs on any microcomputer. I'll attempt to explain the algorithms used in such detail that you will be able to easily transfer the programs into your specific BASIC interpreter. While the ideas for the programs are not necessarily original, I designed and wrote each of the programs from scratch. Thus, they may not reflect the most efficient usage of language, although the astute reader will note an improvement in efficiency in later programs. In other words, my programming got better as I went along.

These are probably some of the most useful applications of microcomputer programs to the home experimenter that have ever been released, especially (as far as I know) in magazine form. I have seen similar listings for sale in book form for as much as $25-$30. I am essentially donating these programs to you by way of this magazine in order to further the art of microcomputer application programming, and in hopes that you will be as unselfish in sharing your machine-level and BASIC language programming with others via the magazines so that we might all ultimately realize the dream of a microcomputer at least in every business, if not in every home.

These programs are designed to run in SWTPC's 6800 8K BASIC. The "no multiple statement lines" problem in SWTPC's interpreter becomes an advantage to those who are trying to convert 6800 BASIC programs to other machines. While you may be able to save on line addresses, you will certainly not have the problem of where to stick a multiple statement line. The programs run in 4K of additional memory. If more memory is available, I would suggest redoing the addresses of the last four programs and calling all these from the Electronic Calculator Executive Routine. This, of course, would be the most convenient procedure. If you have a disk operating system, you can call up the various programs from the disk. One thing I would suggest, however, is that you devote an entire disk to these electronic design programs because there are more coming, and the real, true blue, died-in-the-wool electronics experimenter will find these computer-aided design programs invaluable to him. The programs consist of eleven routines (some of which are called as subroutines from the Executive program). The subroutine programs are: Peak-to-peak to RMS conversion, RMS to peak and peak-to-peak conversion, voltage divider solutions, reactance of inductors and capacitors, inductance of single layer close wound coils, capacitance of parallel plates, and wire tables. Add-on programs, which are not called from the Executive Routine are: Pi-network impedance matching, DBM conversions, Pi-attenuator (DELTA), Tee attenuator (WYE).

Because of the complexity of these programs, this article is laid out in the following order:

(1) General. The general description tells the locations of the programs and describes overall application of each. This provides a quick index of the programs should you have requirements for one of these subroutines without any of the others, or should you desire to load only one or two of the subroutines into your system. You would naturally want the ones that

are most useful to you, and the first part of the article will allow you to determine which is best for you.

(2) Specifics. Here we break down each program or subroutine, and provide a specific description of it. We give the specific formulae used in generating the algorithm and a more detailed description. This part of the article will prove invaluable to those who are rewriting into some limited BASIC or who are rewriting the program for another microcomputer.

(3) Applications. The applications section of this article demonstrates the programs by giving typical runs for each. The algorithms are also described when necessary. This is a problem vs. solution area, wherein a specific problem or task is given and its solution is shown as it will appear on the computer. This is helpful when you're determining which program will be of most use to you as well as in debugging the software. Keep in mind that the various BASIC languages may cause slightly different answers (round off errors, etc.) in some cases. As long as the answers aren't too far out in left field, one can insure that his program has been loaded correctly and is up and running by using the applications section of this article.

Thus, to trace a given program through this article, you should start in the general section, jump to the specifics section, then go directly to the applications.

Every attempt has been made to make my program descriptions, algorithms, flowcharts and formulae as clear as possible. The specific format and layout of the article should allow you to clearly understand what's going on within each of these programs. This is certainly not meant to be a PhD level discussion and I hope it turns out that we have a clear and accurate description of the programs so that you may obtain one of the ultimate applications for a microcomputer system.

General

Refer to Program A for the Electronic Calculator Routines listings.

Peak-to-peak to RMS: The peak-to-peak to RMS program is written as a subroutine called from the electronics calculator executive routine. It is located between steps 20 and 70 in the program listing. It is useful when converting data taken from the screen of an oscilloscope to RMS values. That is, root mean square values which are used for calculations involving power dissipation, etc. Also, the RMS value is the standard value given in circuit specifications.

RMS to peak and peak-to-peak: Statement numbers 70 through 190 contain the RMS to peak and peak-to-peak program which is handled as a subroutine. It may be written as an entire routine with an END statement instead of a RETURN statement, should you decide to use an operating system other than the Executive Routine given in steps 500 through 600. Here you are converting RMS values such as those that will be written in the specifications for a given circuit into the peak and peak-to-peak values which will be seen on the screen of an oscilloscope and ultimately will be used in circuit analysis.

Voltage divider: The voltage divider program is located from statement numbers 200 to 460. Like the others, it may be written as a standalone routine. It solves for any missing part of a classic voltage divider, as shown in Fig. 1. It is useful in designing for a specific voltage drop and allows easy analysis of a voltage divider, especially for transistor biasing. This is very useful in transistor biasing analysis when used in troubleshooting discrete circuits. You may solve for any one item when the other three are given.

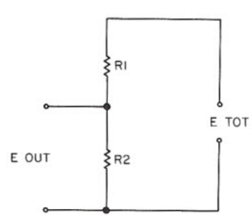


Fig. 1. Voltage divider circuit.

Reactance of L and C circuits: Located in steps 640 through 930, the actual purpose of this program is rather sophisticated. Inputs and outputs are strings. Simply enter as much data as you have on hand, and the program then calculates the rest of the data from the parameters that are available. It will solve for F (resonance) with L and C, and it will also solve for any others such as XC, XL, L or C when F (frequency) and one of the other parameters is given. It is used in both analysis and synthesis; for instance, you may determine the low frequency cutoff in transistor amplifiers or the high frequency limits imposed by circuit board layout (when used with the capacitance of parallel plates program). The program uses pi to the fourth place, 3.1416, and by using pi out as far as possible a higher accuracy is obtainable.

Inductors: Located in steps 1000 through 1095, this short program has a tremendous amount of flexibility. It allows you to solve for N, number of turns; or L, inductance, of a given coil. When solving for number of turns we get involved in circuit synthesis (developing a circuit based upon performance requirements). That is, determining the number of turns that will give a desired inductance in a given circuit. When used in conjunction with the reactance of L and C program we may use this feature to design tuned circuits for oscillators or transmitters.

By using the number of turns in solving for the inductance L, we're involved in circuit analysis and can very closely approximate the inductance of a coil when used in conjunction with the wire tables program or actual measurements of the wires to find the spacing between turns of the coil. By using the inductance program in conjunction with the wire tables program we can also make fairly accurate estimates of coil Q, etc.

Capacitance of parallel plates: Statements 2000 through 3010 contain a program which determines the capacitance of two parallel plates separated by given dielectric. It uses inches or centimeters with centimeters assumed, although inches may be specified. It is highly useful in PC board analysis. When used in conjunction with the capacitive reactance program, one can isolate high frequency loss problems before they occur (in circuit synthesis). This program also allows fabrication of gimmicks (twisted wire capacitors) with reasonable accuracy when used with the wire tables program.

Wire tables: This routine is located in steps 3020 through 3270. It works with copper wire and allows various calculations on wire when the AWG gauge number is input. The first output of the program is wire diameter. When length of wire in feet is entered, the program will give you the weight and resistance calculations for that length of wire. Because of the simplicity of the program in entering and reentering, it may be used for circuit synthesis by using successive approximation to come close to the desired parameters. The ability to calculate in the other direction was not exploited in this program because of the limited abilities of SWTPC 8K BASIC, although an enterprising programmer could give his program these features. It then would be possible to determine the proper AWG gauge when either the diameter, resistance, or weight of a given length was known. Empirical data may be determined from

```
0002 PRINT " "
0005 GOTO 500

0010 REM ELECTRONICS CALCULATOR
0020 PRINT "TO CONVERT P-P TO RMS"
0030 PRINT
0040 PRINT "P-P VOLTS";
0050 INPUT E
0055 PRINT
0060 PRINT .707*(E/2);" RMS"
0070 RETURN

0100 REM RMS TO P AND P-P
0110 PRINT "CONVERT RMS",
0120 PRINT "VALUE";
0130 INPUT R
0140 LET R=R*1.4114
0150 PRINT
0160 PRINT R;" = PEAK"
0170 PRINT 2*R ; " = P-P"
0180 PRINT
0190 RETURN

0200 REM VOLTAGE DIVIDER
0210 PRINT "SOLUTION OF R NETWORK"
0220 PRINT "ENTER R1 OR 0"
0230 INPUT R1
0240 PRINT "ENTER R2 OR 0"
0250 INPUT R2
0260 PRINT TAB (5); "ENTER E TOT OR 0"
0270 INPUT E1
0280 PRINT TAB (5); "ENTER E OUT OR 0"
0290 INPUT E2
0300 IF R1 =0 THEN 350
0305 IF R2=0 THEN 375
0310 IF E1=0 THEN 400
0315 IF E2 <>0 THEN 450
0320 LET X=E1*R2/(R1+R2)
0325 PRINT "THE MISSING NUMBER IS ";X
0330 PRINT
0335 RETURN
0350 LET X=R2*(E1-E2)/E2
0360 GOTO 325
0375 LET X=(E2*R1)/(E1-E2)
0380 GOTO 325
0400 LET X=(E2*(R1+R2))/R2
0410 GOTO 325
0450 PRINT "TOTAL RESISTANCE IS ";R1+R2; " OHMS"
0460 RETURN
0500 REM ELECTRONICS CALC
0510 PRINT TAB (8); "WHICH PGM"
0520 PRINT "1. RESISTORS"
0530 PRINT "2. P-P TO RMS"
0540 PRINT "3. RMS TO P & P-P"
0545 PRINT "4. AC CKTS"
0546 PRINT "5. SINGLE LAYER COIL"
0547 PRINT "6. CAP OF PARA PLATES"
0548 PRINT "7. WIRE TABLES"
0550 INPUT C
0555 IF C=4 GOSUB 640
0560 IF C=1 GOSUB 200
0570 IF C=2 GOSUB 10
0580 IF C=3 GOSUB 100
0581 IF C=5 GOSUB 1000
0582 IF C=6 GOSUB 2000
0583 IF C=7 GOSUB 3020
0585 INPUT "ENTER 0 TO QUIT",F
0586 IF F=0 END
0590 PRINT " "
0600 GOTO 500
0640 PRINT " "
0642 REM RLC PGM******
0650 PRINT "ENTER FLC XC OR XL THEN COMMA"
0660 PRINT "THEN ENTER NUMERIC DATA"
0670 PRINT TAB (8) ;"CALC,0 STARTS PGM"

0680 PRINT "ENTRY";
0700 INPUT A$,A
0710 IF A$="F" THEN LET F=A
0720 IF A$="C" THEN LET C=A
0730 IF A$="L" THEN LET L=A
0740 IF A$="XC" THEN LET X1=A
0745 IF A$="XL" THEN LET X2=A
0750 IF A$="CALC" THEN GOTO 800
0755 GOTO 700
0800 PRINT "DESIRED VALUE (X,F,C,OR L)";
```

"EXECUTIVE
Routine

```
0805 INPUT A$
0810 IF ASC(A$)=88 GOTO 850
0815 IF ASC(A$)=67 GOTO 875
0817 IF ASC(A$)=76 GOTO 900
0820 LET F=1/((SQR(C*L))*6. 2832)
0825 PRINT A$;" = ";F
0830 RETURN
0850 IF A$="XC" GOTO 925
0860 LET F=6. 2832*L*F
0870 GOTO 825
0875 LET F=1/(6.2832*F*X1)
0880 GOTO 825
0900 LET Y=X2/6. 2832*F
0910 LET F=Y
0920 GOTO 825
0925 LET F=1/(6. 2832*C*F)
0930 GOTO 825
1000 REM INDUCTANCE OF SINGLE LAYER COIL

1010 REM WRITTEN BY JIM HUFFMAN
1020 REM 2/7/77
1030 DIGITS= 2
1040 GOSUB 1096
1050 INPUT "NO. TURNS",N
1060 INPUT "RADIUS",R
1070 INPUT "SPACING",D
1075 INPUT "INDUCTANCE DESIRED (UH)",L
1077 IF N=0 THEN 1082
1080   L=(N*N*R*R)/(9*R+10*N*D)
1082   X=(5*D*L)/(R*R)
1084   Y=SQR((9/R)*L)
1086   N=SQR((X*X)+(Y*Y))
1087   N=N+X
1089 PRINT
1090 PRINT TAB(10);"INDUCT =";L;"UH"
1092 PRINT TAB(10);"NO. TURNS =";N
1095 RETURN

1096 PRINT " "
1097 RETURN

2000 REM CAPACITANCE OF PAR PLATES
2010 REM WRITTEN BY J HUFFMAN
2020 REM 2/7/77
2030 GOSUB 1096
2035 DIGITS= 3
2040 INPUT "PERMITTIVITY OF INSULATION", E
2050 INPUT "DISTANCE BETWEEN PLATES COMMA IN OR CM",D,D$
2055 IF D$="IN" THEN D=D*2. 54
2060 INPUT "LENGTH OF PLATES COMMA IN OR CM",L,L$
2065 IF L$="IN" THEN L=L*2.54
2070 INPUT "WIDTH OF PLATES COMMA IN OR CM",W,W$
2075 IF W$="IN" THEN W=W*2.54
2080   X=D/(3.1416*W)
2085   X=X*(1+LOG((6.2832*W)/D))
2087 IF 100*W >L THEN X=0
2090   C=. 0885419*((E*L*W)/D)
2095   C=C*(1+X)
3000 PRINT "CAPACITANCE = ";C;" PF"
3010 RETURN
3020 REM ****WIRE TABLES ***
3030 REM BY J HUFFMAN
3040 REM WRITTEN 1/26/77
3050 REM ENTRY ROUTINE
3060 GOSUB 1096
3070 PRINT TAB(8); "WIRE CALCULATIONS"
3080 PRINT
3090 DIGITS= 3
3100 REM DIA CALCULATIONS****
3110 REM
3120 INPUT "AWG GAUGE",G
3130   X=(G+3)/39
3140   D=(460/(92↑X))
3150 PRINT TAB(8); "DIAMETER =";D;" MILS"
3200 REM RESIST CALCULATIONS
3210 INPUT "NO. OF FT",N
3220   R=(10371/(D*D))/(1000/N)
3230 PRINT TAB(8); "RESISTANCE =";R;" OHMS"
3240 REM WEIGHT CALCULATIONS**
3250   W=(.0030269*(D*D))/1000/N)
3260 PRINT TAB(8); "WEIGHT =";W;" LBS"
3270 RETURN
```

Program A. Electronics Calculator Routines.

this program by using a length of 1000 feet. Thus, you can determine the weight per 1000 feet for estimating shipping costs for an electronics business.

Pi-network impedance matching: This is the first of the routines that is not written as a subroutine and cannot be called from the executive (Electronics Calculator) program. It begins at statement 1 and continues through 650, allowing the synthesis or analysis of pi-network circuits. Program B is a listing of the routine. Although, as in the wire tables program, analysis must be done by successive approximations, the pi-network impedance matching program is most useful in designing rf

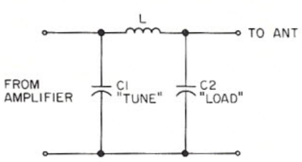


Fig. 2. The classic pi-network circuit (being used to match the output amplifier of a radio transmitter to the antenna).

amplifiers. The parameters required by the program are desired Q (measure of bandwidth of the tuned circuits); R1, output impedance of the stage or input impedance of the network; and R2, the impedance to be matched. Fig. 2 shows the classic configuration that this program solves for. Also the computer draws a limited graphics schematic, as shown in the applications section.

DBM conversions: Compared to the rest of the programs, this program is rather lengthy, but it has extreme flexibility. It is written as a standalone routine in statements 10 through 1410 (see Program C). It is fairly complex, but determines resistance, DBM, or volts, given the other two inputs. This is an invaluable design aid when working with microphones or phonographic pickups and audio frequency design. It also allows analysis and design of telephone equipment systems and it is unlimited in its capabilities. For instance, it can be used to determine the approximate output voltage of a microphone specified as a -52 dB output with a 50K load, and could be just as easily used to determine the DBM levels of a .7 volt RMS signal into a 600 Ohm load.

Pi attenuator program: The pi attenuator program works in conjunction with the tee attenuator program. It is another standalone program and goes from statement 5 through 520 (see Program D). There is a main routine shared by the two programs at 600 through 690. By inspection of steps 120 through 150, one can see that the calculations for this program are rather complex. The pi attenuator is also known as the delta attenuator. The program supplies missing resistance values and should not be confused with the pi-network impedance matching program which uses inductor capacitor values for circuit synthesis. The program also works for impedance matching because it supplies minimum loss data.

Tee attenuator: The tee or wye attenuator operates on basically the same principles as the pi attenuator program and works with the pi attenuator program, in that an Executive Routine is shared by both programs. This allows evaluation of both tee and pi attenuator programs to determine optimum values in given circuit conditions.

Program Specifics

Electronic Calculator Routines: In steps 500 through 600 the processor prints a selection of seven programs that may be chosen. In steps 510 through 548, depending on which program is chosen, the processor then goes to a given subroutine. On its return, it will prompt with "ENTER 0 TO QUIT", and if the input equals zero, the Executive Routine ends. Otherwise, you go back to the beginning of the Executive Routine, print out the selections, and call up whichever one is needed in that desired program. This is the Electronics Calculator Executive Routine and it can be replaced in disk operating systems by calling programs from the disk operating system, when each of the subroutines called by the Executive Routine are treated as standalone routines, as already discussed.

Peak-to-peak to RMS: Peak-to-peak to RMS program is the first of the Electronics Calculator Routines. We can determine how it works by looking at steps 10 through 70 of the program listing. After the message CONVERT PEAK-TO-PEAK TO RMS is printed you input the desired peak-to-peak voltage to be converted to RMS. If you were using peak voltage and needed to convert that to RMS, you'd merely multiply it by two in your head and enter that value at step 50. Step 60 then prints the RMS value. Dividing the peak-to-peak voltage by two, you'd receive peak, and multiplying times $1/\sqrt{2}$ will determine the Root Mean Square value. At 70 the program contains a return, but if the program were to be written as a standalone program such as the one used in a disk operating system, step 70 would be an END statement. The formula used in this program is quite simple and is given at step 60; therefore, it won't be repeated.

RMS to peak and peak-to-peak: Here at statement 110, our processor prints "CONVERT RMS" then queries for value R in step 120 and 130. Step 140 calculates the peak value by multiplying the RMS value input at step 130 times $\sqrt{2}$ then printing the value at step 160. Step 170 multiplies the peak value by 2 to determine the peak-to-peak value and in step 190 the program returns to the calling routine. Keep in mind that step 190 can be an END statement and the subroutine becomes a standalone routine.

Voltage divider: The voltage divider subroutine begins at step 200. Step 210 merely prints a heading while 220 and 230 are used to input value of resistor R1. At this point, either the value of R1 or a zero is entered. Steps 240 and 250 do the same for R2. Steps 260 and 270 are used to enter the total voltage across the divider which is handled in the program as E-E1. Steps 280 and 290 process E2 in the same way. In either of the preceding steps a zero could be entered, if the values were unknown. Steps 300 through 315 are used to determine which of the values is missing and which must be solved for. Take special note of step number 315. If the program has looked at each value and gets to step 315 with E2 not equal to zero, then you

```
0002 REM BY JR HUFFMAN
0003 REM JAN 1977
0005 DIGITS= 3
0010 PRINT " "

0020 INPUT "R1",R1
0030 INPUT "R2",R2
0040 INPUT "FREQ",F
0050 INPUT "Q",Q
0060 IF Q<SQR(R1/R2-1) THEN 600
0070 GOTO 500

0100 PRINT "O ----- UUUUU -----O"
0110 PRINT "    I      L=";L
0120 PRINT "    I                    I"
0130 PRINT "   ---                  ---"
0140 PRINT "C1=";C1;"          C2=";C2
0150 PRINT "   ---                  ---"
0160 PRINT "    I                    I"
0170 PRINT "    I                    I"
0180 PRINT "O --------------------O"
0190 RETURN

0500  X1=R1/Q
0510  C1=1/(2*3.1416*X1*F)
0520  X2=((R2/R1)*(Q*Q+1))-1
0530  X2=R2/(SQR(X2))
0540  C2=1/(2*3. 1416*F*X2)
0550  X3=(Q*R1)/(Q*Q+1)
0560  X3=X3*((R2/(Q*X2))+1)
0570  L=X3/(2*3. 1416*F)
0580 GOSUB 100
0590 PRINT
0595 END

0600 PRINT " "
0610 PRINT
0620 PRINT "INVALID PARAMETER — Q<SQR(R1/R2-1)"
0630 PRINT
0640 PRINT "RE-ENTER ";
0650 GOTO 20
```

Program B. Pi-network impedance matching.

```
0010 REM ****DBM CONVERSIONS**
0020 REM BY JIM HUFFMAN
0030 REM 1/21/77
0040 REM

0050 REM INPUTDATA********
0060 REM
0065 GOSUB 1400
0070 PRINT "VOLTAGE TO DBM PROGRAM. . . .WANT INSTRUCTIONS";
0080 INPUT A$
0090 IF A$="YES"THEN GOSUB 1000
0100 IF A$="Y" THEN GOSUB 1000
0110 GOSUB 1400
0111 INPUT "IMPEDANCE IN OHMS",I
0120 INPUT "VOLTAGE IN V RMS",V
0130 INPUT "LEVEL IN DBM ",D

0140 REM SOLUTIONS******
0150 REM
0160 PRINT
0170 INPUT "SOLVE FOR (DBM, VOLTS, IMP)",S$
0180 IF ASC(S$)=68 THEN 200
0190 IF ASC(S$)=86 THEN 300
0192 IF ASC(S$)=73 THEN 400
0195 PRINT "   . . . . ERROR. . ."
0197 GOTO 160
0200 REM **SOLUUTION FOR DBM***
0210 REM
0220 LET X=V*V
0230 LET Y=(LOG(X/I)/LOG(10))
0240 LET X=10*(Y+3)
0250 PRINT "SOLUTION:"
0260 PRINT ". . . . .DBM=";X
0270 PRINT
0280 PRINT "VOLTS = ";V
0290 PRINT "IMP = ";I
0295 INPUT "ANOTHER (1=YES,0=NO)",A
0297 IF A=1GOTO110
0298 END

0300 REM **SOLUTION FOR VOLTS**
0310 REM
0320  X=(D-30)/10
0330  Y=10↑X
0340  X=SQR(Y*I)
0350 PRINT "SOLUTION:"
0360 PRINT ". . . . .VOLTAGE =";X;" RMS"
0370 PRINT
0380 PRINT "IMPEDANCE=";I
0390 PRINT " DBM =     ";D
0392 GOTO 295

0400 REM *SOL FOR IMPEDANCE**
0410 REM
0420  X=(30-D)/10
0430  Y=(10↑X)*V*V
0440 PRINT "SOLUTION :"
0450 PRINT ". . . .IMPEDANCE=";Y;"OHMS"
0460 PRINT
0470 PRINT "VOLTS=";V
0480 PRINT "DBM = ";D
0490 GOTO 295

1000 REM INSTRUCTIONS PRINTOUT**
1010 REM
1015 LINE= 120
1020 GOSUB 1400
1030 PRINT " THIS PROGRAM SUPPLIES THE UNKNOWN QUANTITY";
1040 PRINT "WHEN ANY TWO VALUES ARE GIVEN."
1050 PRINT
1060 PRINT " ENTER KNOWN VALUES WITH THE VALUE AND THE ";
1070 PRINT "UNKNOWNS AS 0 OR 1"
1080 PRINT "THE PROGRAM WILL PROMPT FOR SOLUTION ";
1090 PRINT "OF THE UNKNOWN"
1095 LINE= 40
1097 INPUT "FINISHED? ENTER C/R",A$
1100 RETURN

1400 PRINT "  "
1410 RETURN
```

Program C. DBM Conversions.

go to step 450 which merely adds the resistance of R1 and R2 and prints it out, because there were no missing values entered into the program. Step 320 is selected by default; i.e., E2 must be equal to zero, and contains the first of the mathematical formulas. Steps 325, 330 and 335 are shared by all the routines in that X is used as the unknown number; i.e., in step 300, if R1 = 0, then you go to step 350 in which you let X equal the value shown. Step 360 then sends you to step 325 which prints THE MISSING NUMBER IS X — in this case, R1. The same thing occurs in steps 375 and 400. The formulas used in the solution of this program are based on the formula given in Fig. 3 and all variations are merely solutions of this general formula.

$$\frac{\text{ETOT}}{\text{R1 + R2}} = \frac{\text{EOUT}}{\text{R2}}$$

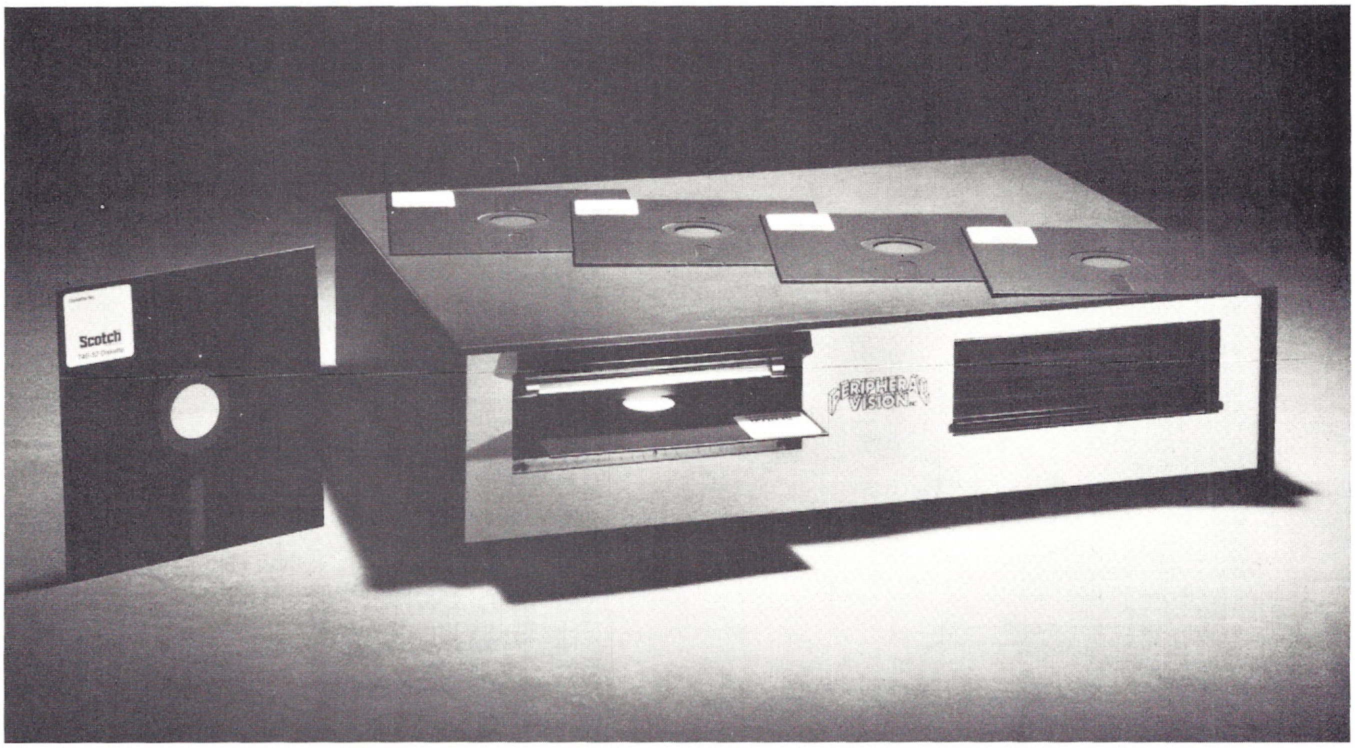
Fig. 3. Formula for voltage divider calculations.

Reactance of L and C: Note step 640. Here we print control characters which control homeup and erase-the-screen on my CT1024 terminal. This would obviously be replaced by whatever is necessary to clear the screen on your given display system. Instructions in step 650 tell you to enter F, L, C, XC or XL, then a comma, then enter the numeric data. Once all the known values are entered, to start up the program, you enter "CALC", then "," then "0" as the numeric value. On hitting carriage return, the program begins. Note in steps 710 through 750 that each of the input string statements are inspected. As soon as the string value = CALC, then you go to step 800 and the processor prompts for the desired value — that is X, F, C or L. Again we put in A$. This time, we convert the ASCII value of the string, or inspect to see whether we are solving for X, F, C or L. Again, all the programs are sharing steps 825 and 830. Thus, the solution of all the programs is for F. At step 820, which is selected by default, F is solved when C and L are given by using the classic formula

$$F = \frac{1}{2\pi \cdot \sqrt{LC}}$$

Again in 825, we merely print the A$ or whatever desired value is input. In this case, F = F; then the actual numeric value of F is determined in the formula, and we return to the calling routine in step 830. In step 850 we have determined that the first ASCII character in the string is an X. This has sent us to step 850. In step 850 we decide whether or not we are solving for XC or XL. If A$ is equal to XC, we are sent to step 925 — if not, we got to

FLIP OVER OUR FLOPPY



Only $750 from Peripheral Vision.

Peripheral Vision is a brand-new company that's dedicated to selling reasonably priced peripherals for various manufacturers' CPU's.

We think you'll flip over our first product.

It's a full-size floppy disk for the Altair-Imsai plug-in compatible S-100 BUS. And it's available for as low as $750.

Here are the features:

- 1 interface card supports 4 drives
- Stores over 300,000 bytes per floppy
- Bootstrap EPROM included—no more toggling or paper tape
- Completely S-100 plug-in compatible
- Interface cabling included
- Drive is from Innovex (the originator of the floppy concept)—assembled and tested
- Interface card design is licensed from Dr. Kenneth Welles and the Digital Group
- Disk operating system with file management system included on floppy
- Cabinet and power supply optional

Prices:	Kit	Assm.
Interface card kit and assembled and tested drive	$750	$850
Power supply— +24V at 2A	45	65
Cabinet—Optima, blue	—	85

Now, a little more about our company.

Peripheral Vision may be brand-new, but we have some old-fashioned ideas about how to run our business.

We know there are serious incompatibilities among the different manufacturers' peripherals and CPU's. We want to get them together. And, we want to bring significant new products to market—products consisting of everything from adaptation instructions/kits for hardware and software to major new products.

It's a tall order, but we feel we're up to the task. Peripheral Vision has already obtained a license from The Digital Group to adapt versions of some of their products to the S-100 BUS. And we're working on getting more from other companies.

Most important to our customers, Peripheral Vision is committed to helping you get along with your computer. We'll do all we can to make it easy.

Write us now for all the information on our company, our philosophy and our exciting line of products. And be prepared to flip over all of it.

PERIPHERAL VISION INC. P12

P.O. Box 6267/Denver, Colorado 80206/(303) 733-1678

Send me the works, and I just might flip over it!

Name____________________

Address____________________

City/State/Zip____________________

```
0005 GOTO 600
0010 REM *****PI ATTENUATOR
0020 REM BY JIM HUFFMAN
0030 REM 1/20/1977
0040 REM

0050 REM MAIN ROUTINE ******
0060 GOSUB 1400
0065 PRINT TAB(4); "*** PI NET ATTEN ***"

0070 INPUT "INPUT IMPEDANCE",I
0080 INPUT "OUTPUT IMPEDANCE",O
0090 INPUT "LOSS IN DB",L
0100 REM COMPUTE STUFF *****
0110 LET L=ABS(L)
0115 LET L=L/10
0117 LET L=10↑L
0120  R3=(L-1)*.5*SQR(I*O/L)
0130  X=(L+1)/(L-1)
0132  R1=1/(((1/I)*X)-(1/R3))
0140  R2=1/(((1/O)*X)-(1/R3))
0145  X=SQR(I/O)
0146  Y=SQR((I/O)-1)
0147  X=(X+Y)*(X+Y)
0148  M=10*LOG(X)/LOG(10)
0150 REM DISPLAY RESULTS ********
0155 IF SGN(R1)=-1 THEN 500
0160 GOSUB 1400
0170 PRINT "O - - I - - - R3 - - - I - - O"
0180 PRINT "     I              I"
0190 PRINT "     R1            R2"
0200 PRINT "     I              I"
0210 PRINT "O - - I - - - - - - - - - - I - - O"
0220 PRINT
0230 PRINT "R1=";R1
0240 PRINT "R2=";R2
0250 PRINT "R3=";R3
0260 PRINT
0270 PRINT "MIN LOSS  = ";M
0275 INPUT "NOW HIT C/R",C$
0280 RETURN
0500 PRINT "DESIRED LOSS ";L; "TIMES. ."
0505 PRINT "IS MORE THAN MIN LOSS OF ";M; "DB"
0510 PRINT "REENTER"
0520 GOTO 70

0600 GOSUB 1400
0601 REM +++++MAIN ROUTINE
0605 PRINT TAB(4); "***************"
0610 PRINT "ATTENUATOR DESIGN PGM"
0620 PRINT
0630 PRINT "CHOOSE:"
0640 PRINT
0650 PRINT "     1. TEE NET ATTEN"
0655 PRINT "     2. PI NET ATTEN"
0660 INPUT C
0670 IF C=1 GOSUB1000
```

```
0680 IF C=2 GOSUB 10
0690 GOTO 600
1000 REM ****T ATTENUATOR PGM
1030 REM

1040 REM INPUT*******
1045 GOSUB 1400
1047 PRINT TAB(4); "*** TEE NET ATTEN ***"
1060 INPUT "INPUT IMPEDANCE",I
1070 INPUT "OUTPUT IMPEDANCE",O
1080 INPUT "DESIRED LOSS -DB",L
1090 GOSUB 1200
1095 IF SGN(R2)=-1 THEN 1300

1100 REM PRINTOUT ROUTINE*****
1110 REM
1115 GOSUB 1400
1120 PRINT "O - - - R1 - - I - - R2 - - - O"
1130 PRINT "               I"
1140 PRINT "               I"
1150 PRINT "               R3"
1160 PRINT "               I"
1170 PRINT "               I"
1180 PRINT "O - - - - - - - I - - - - - - - O"
1182 PRINT TAB(4);"R1=";R1
1184 PRINT TAB(4);"R2=";R2
1186 PRINT TAB(4);"R3=";R3
1188 PRINT
1190 PRINT "MINIMUM LOSS=";X;" DB."
1191 INPUT "NEXT HIT C/R",C$
1195 RETURN

1200 REM CALCULATIONS SUBR******
1210 REM
1215  L=L/10
1216  L=10↑L
1220  R3=2*(SQR(L*I*O))/(L-1)
1230  R2=O*((L+1)/(L-1))-R3
1240  R1=I*((L+1)/(L-1))-R3
1250  X=(SQR(I/O)+SQR((I/O)-1))
1260  X=X*X
1270  X=LOG(X)/LOG(10)
1280  X=10*X
1290 RETURN

1300 REM LOSS PROBLEM****
1310 LINE= 64
1315 GOSUB 1400
1320 PRINT "DESIRED LOSS ";L;"TIMES.";
1330 PRINT "GREATER THAN MIN LOSS ";X;"DB";
1340 PRINT " THUS YOU MUST REENTER DATA"

1350 PRINT
1360 GOTO 1060

1390 REM ******H/U AND CLEAR
1400 PRINT " "
1410 RETURN
```

Program D. Pi and tee attenuator routines.

step 860 by default and let F equal, in this case, XL. After determining the value of XL by using the formula, XL = 2π FL. We are then sent in step 870 to step 825 where we print A$ (which was XL) equals F; then RETURN. In 850, had A$ equaled XC, we would have been sent to step 925 which uses the formula

$$XC = \frac{1}{2\pi\ FC}$$

In step 930 we are returning to step 825 which causes printing of A$ or XC = F. In step 815, if we are solving for C, we are sent to step 875 which uses the formula

$$C = \frac{1}{2\pi\ FX_C}$$

(X1 in this case; see step 740). Then go to step 825 for a printout of our answer. At step 817, if we are looking for L we are sent to step 900. Here the formula used is

$$L = \frac{X_L}{2\pi\ F}$$

We use Y as our unknown, and then in step 910 let F = Y, go to 825 and print the answer.

Inductance of a single layer close wound coil: Here a step by step analysis of the program shows that at step 1030 command DIGITS = 2 truncates our answer to the nearest two decimal points. Step 1030 is not absolutely necessary to the program, but is a useful feature in the SWTPC 8K BASIC. Subroutine call is at step 1040. Once again, subroutine 1096 causes homeup and erase of the cursor on my CT1024 terminal. This subroutine would have to be adapted to your particular terminal to clear the screen off. In my case, it outputs a control P followed by a control V. At step 1050, an input statement is used. Although, if your BASIC is somewhat limited, you may have to insert a statement such as PRINT "NUMBER TURNS" followed by the input statement INPUT , N. Step 1060 allows input of radius R; 1070, input of spacing D. Step 1075 is the point where you input the inductance required. At 1050 or 1075, you may insert a zero, although radius R and spacing D will always have to be given. The radius R is the radius of the coil in inches; the spacing D is determined by the diameter of the wire and is, in effect, the same as the diameter of the wire when the coil is close wound. It is, in effect, the center to center spacing between turns. Note in step 1077, if the

number of turns = 0, then we are to solve for the number of turns and are sent to step 1082. Here, in effect, we skip step 1080 which is the solution for inductance L, and we go through the rest of the program. Were the number of turns given, and the inductance L the desired product, then by default we would elect to go to step 1080 and determine inductance L. The program would also determine N number of turns. Since N number of turns equals input anyway, this will make no difference. The output from the computer as provided in steps 1090 and 1092 will be both inductance L and number of turns N under any missing value conditions. The formula used for deriving the other formulas in this program is shown in Fig. 4a.

Capacitance of parallel plates: In step 2035 a three digit to the right of the decimal point readout is selected and data begins to be input at step 2040. When the computer prompts for the permittivity of the insulation, this is comparable to the dielectric constant. Step 2050 calls for a distance between plates followed by specifying whether or not the input is inches (in) or centimeters (cm). D is distance and D$ specifies inches or centimeters. Step 2055 then operates on D$. If D$ specifies inches, then the value of D is multiplied times 2.54, converting it to centimeters. If the input is entered in centimeters the program will compute directly. All you must do is enter the distance between plates D and then hit carriage return. By default, centimeters will be selected.

At step 2060, the computer is prompting for the length of plates separated by a comma from L$ whether or not the value given for L is in inches or centimeters. Querying each step as it is taken allows intermixing of inch and centimeter measurements. The mathematics occur in steps 2080 through 2095. Step 3000 gives the printout of capacitance C in picofarads. The formula used for solving the capacitance of two parallel plates is given in Fig. 4b.

Wire tables: Step 3070 prints a heading WIRE CALCULATIONS and 3090 is used to truncate to three digits to the right of the decimal point. At step 3120, the computer prompts for the AWG gauge number G, and at steps 3130 through 3140 you solve for diameter D, by using the formula shown in Fig. 4c. The answer is printed in step 3150 as diameter D in mils. The number of feet is prompted for in step 3210. Both the resistance (step 3220) and weight (step 3250) are calculated and then printed out at steps 3230 and 3260, respectively. Resistance and weight calculations are also given in Fig. 4c. Following are descriptions of the three standalone routines.

Pi-network impedance matching design program: Uses the formulae given in Fig. 5 for determination of its values. The program listing is shown in Program B. Step 5 truncates the answer to three digits. Step 20 prompts for R1. Steps 30, 40 and 50 prompt for R2, frequency and Q, desired respectively. If the value of Q is too low, you are sent to a trap in step 600 where first the screen is cleared and at step 620 the invalid parameter message is written on the screen. Step 640 tells you to reenter and step 650 sends you back to the beginning and calls for new values of R1, R2, F and Q. Step number 70 sends the program to step 500 and steps 500 through 570 are where the calculations take place. In step 580 you are sent to subroutine 100 and in steps 100 through 190, you print the schematic diagram shown, filling in the values for L, C1 and C2.

DBM conversions: Refer to Program C for the following discussion. At step 50 data is input. Step 111 prompts for input impedance in Ohms; step 120, voltages in volts RMS; step 130, the level in DBM. By entering zero or one for any of these you may solve for the others. Since zero input is a valid DBM input at step 130, you may enter one for DBM. Step 170 prompts for the solution by inspecting the first character of the input string for D, V or I. If none of these are entered, step 195 will cause the printing of an error message and step 197 will send you back to prompt for DBMs, volts, or impedance. In step 180, if the first character of the string value is D, you will be sent to step 200 which is the solution for DBM. Mathematical processes are contained in step 220 through 240, and at step 250 the computer prints the solution. Step 260, DBM equals the value that was solved for. Step 280 prints the other two variables; step 295 prompts for another input, and step 297 makes the decision whether to go back to 110 and work another problem, or to end the program, as given at step 298. Back at step 190, if the input solution calls for voltage V, then you are sent to step 300. The mathematics is performed in steps 320 through 340. Step 350 prints solution; 360, voltage is solved; 380 and 390 give the other two variables and step 392 sends you to step 295 where you are prompted to see if you wish to solve another problem or end the program as already discussed. Step 400 is the solution for impedance; mathematics is contained in steps 420 and 430; solution printout and printout of the other variables is in steps 440 through 480. Again, at step 490 you are sent back to step 295 where you are prompted

g = gage of wire in mils
R = radius of coil
N = number of turns
D = diameter
L = inductance (μH)

(a) $$L = \frac{N^2R^2}{gR + 10\,ND}$$

(b) $$C = .0885419\,\frac{\epsilon_r\,LW}{d}\,(1 + 0^*)$$

$$^*\ 0 \text{ when } 100\,W > L \text{ or } \frac{d}{\pi W}\left(1 + \ln \frac{2\pi W}{d}\right) \text{ when } L \geqslant 100W$$

ϵ_r = permittivity of insulator
d = thickness of insulator
L = length of plates (cm)
W = width of plates (cm)
C = capacitance (pF)
Formula most accurate when L >> d and W >> d.

(c) $$\text{Dia} = \frac{460}{92^{\frac{AWG+3}{39}}}$$

$$R = \frac{10371}{\text{Dia}^2}$$

$$W = .0030269\,(\text{Dia}^2)$$

Fig. 4. (a) Inductance formula. (b) Capacitance of parallel plates formula. (c) Wire diameter, weight (W) and resistance (R) formulae.

to see if you wish to continue with other data or to exit the routine.

Pi and tee attenuator programs: Refer to Fig. 6 for the formulae for these two routines and Program D for the listings. Step 5 sends you to step 600 which is the main routine that ties both programs together. In step 610 the heading ATTENUATOR DESIGN PROGRAM is printed. In 630, you prompt for a choice of 650 or 655, either one, tee net attenuator program, or to pi-net attenuator program. At step 670, if you have chosen the tee-net attenuator program, you are sent to subroutine 1000; if you have chosen number two, you will be sent to subroutine 10. Step 690 puts you back in to the main routine at step 600 again. We'll begin by discussing subroutine 10 or the pi-net attenuator program, which is broken into a main program and a display results program. At step number 65, the heading PI-NET ATTENUATOR is printed on the screen. At steps 70, 80 and 90, you are prompted for input impedance, output impedance and loss. Steps 100 through 148 are used to compute values of the three resistors in the pi attenuator program and correspond to the formulae given in Fig. 6. At step 150, you will begin to display the results. At 155, if the sign of R1 is negative, then you must go to step 500, which says desired loss L is more than the minimum loss of M in decibels. You will then go to step 510 which says reenter, and step 520, which sends you to step 70 to reenter input impedance, output impedance and desired loss in dB. In step 155, if the sign of R1 is not equal to -1, then the calculations are valid and steps 160 through 220 will print a pictorial diagram of the positions of R1, R2, and R3. Steps 230 through 250 will print the values of R1, R2, and R3. At step 270, the processor will print the value of the minimum loss, M. Step 275 prompts for a carriage return if you are finished transposing the data from the CRT screen. At step 280, upon receiving the carriage return, the program will return to the calling routine which is the main routine, back at 690.

The tee attenuator program begins at step 1000; clears the screen in step 1045; prints a heading in 1047; prompts for input impedance, output impedance and desired loss in steps 160, 170 and 180. Step 190 then sends you to subroutine 1200 which performs the mathematical calculations based on the formulae given in Fig. 6. If the sign of R2 = -1, then you go to step 1300 which

$$X_{C1} = \frac{R1}{Q}$$

$$C_1 = \frac{1}{2\pi F X_{C1}}$$

$$X_{C2} = \frac{R2}{R1\,(Q^2+1)-1}$$

$$C2 = \frac{1}{2\pi F X_{C2}}$$

$$X_{L1} \quad \left(\frac{QR1}{Q^2+1}\right) \quad \left(\frac{1+R2}{Q\,X_{C2}}\right)$$

$$L_1 = \frac{X_{L1}}{2\pi F}$$

Fig. 5. Formulae for pi-network impedance matching.

(a)

$$R_3 = \frac{1}{Z}\,(N-1)_1\,\left(\frac{Z_1\,Z_2}{N}\right)^{1/2}$$

$$R_1 = \frac{1}{Z_1}\left(\frac{N+1}{N-1}\right) - \frac{1}{R_3}$$

$$R_2 = \frac{1}{\frac{1}{Z_2}\left(\frac{N+1}{N-1}\right) - \frac{1}{R_3}}$$

$$\text{Min Loss} = \frac{10 \quad \log_N \left(\sqrt{\frac{Z_1}{Z_2}} + \sqrt{\frac{Z_1}{Z_2} - 1} \right)^2}{\log_N (10)}$$

$$\text{Loss Desired} = 10 \log_{10} N$$

(b)

$$R_3 = \frac{2\sqrt{N\,Z_1\,Z_2}}{N-1}$$

$$R_1 = Z_1\left(\frac{N+1}{N-1}\right) - R_3$$

$$R_2 = Z_2\left(\frac{N+1}{N-1}\right) - R_3$$

$$\text{Min Loss} = 10 \log_{10} \left(\sqrt{\frac{Z_1}{Z_2}} + \sqrt{\frac{Z_1}{Z_2} - 1} \right)^2$$

$$\text{Loss Desired} = 10 \log_{10} N$$

Fig. 6. Math for (a) pi-network attenuator, (b) tee-network attenuator.

```
#RUN

      WHICH PGM
1. RESISTORS
2. P-P TO RMS
3. RMS TO P & P-P
4. AC CKTS
5. SINGLE LAYER COIL
6. CAP OF PARA PLATES
7. WIRE TABLES
? 1
SOLUTION OF R NETWORK
ENTER R1 OR 0

? 0
ENTER R2 OR 0
? 1000
ENTER E TOT 00
? 12
ENTER E OUT OR 0
? 3
THE MISSING NUMBER IS 3000

ENTER 0 TO QUIT? 1

      WHICH PGM
1. RESISTORS
2. P-P TO RMS
3. RMS TO P & P-P
4. AC CKTS
5. SINGLE LAYER COIL
6. CAP OF PARA PLATES
7. WIRE TABLES
? 2
TO CONVERT P-P TO RMS

P-P VOLTS? 3.67

1.297345 RMS
ENTER 0 TO QUIT? 2

      WHICH PGM
1. RESISTORS
2. P-P TO RMS
3. RMS TO P & P-P
4. AC CKTS
5. SINGLE LAYER COIL
6. CAP OF PARA PLATES
7. WIRE TABLES
? 3
CONVERT RMS
VALUE? 115
162.311 = PEAK
324.622 = P-P

ENTER 0 TO QUIT? 3

      WHICH PGM
1. RESISTORS
2. P-P TO RMS
3. RMS TO P & P-P
4. AC CKTS
5. SINGLE LAYER COIL
6. CAP OF PARA PLATES
7. WIRE TABLES
? 4
ENTER FLC XC OR XL THEN COMMA
THEN ENTER NUMERIC DATA
   CALC,0 STARTS PGM
ENTRY? F,10000
? C,.001E-6
? CALC,0
DESIRED VALUE (X,F,C,OR L)? XC
XC = 15915.457
ENTER 0 TO QUIT? 1

      WHICH PGM
1. RESISTORS
2. P-P TO RMS
3. RMS TO P & P-P
4. AC CKTS
5. SINGLE LAYER COIL
6. CAP OF PARA PLATES
7. WIRE TABLES
? 5

NO. TURNS? 0
RADIUS? .25
SPACING? .036
INDUCTANCE DESIRED (UH)? 10

   INDUCT =10.00 UH
   NO. TURNS =63.28
ENTER 0 TO QUIT? 1

   WHICH PGM
1. RESISTORS
2. P-P TO RMS
3. RMS TO P & P-P
4. AC CKTS
5. SINGLE LAYER COIL
6. CAP OF PARA PLATES
7. WIRE TABLES
? 6

PERMITTIVITY OF INSULATION? 4.5
DISTANCE BETWEEN PLATES COMMA IN OR CM? .065,IN
LENGTH OF PLATES COMMA IN OR CM? 10
WIDTH OF PLATES COMMA IN OR CM? .005,IN
CAPACITANCE = 0.652 PF
ENTER 0 TO QUIT? 1

   WHICH PGM
1. RESISTORS
2. P-P TO RMS
3. RMS TO P & P-P
4. AC CKTS
5. SINGLE LAYER COIL
6. CAP OF PARA PLATES
7. WIRE TABLES
? 7

   WIRE CALCULATIONS

AWG GAUGE? 28
   DIAMETER =12.641 MILS
NO. OF FT? 123
   RESISTANCE =7.982 OHMS
   WEIGHT =0.0594 LBS
ENTER 0 TO QUIT? 0

READY
#
```

Example A. Applications/sample RUNs of Electronics Calculator Programs.

```
#RUN

R1? 125
R2? 50
FREQ? 28E6
Q? 3
O - - - - - UUUUU - - - - - O
      I    L=3.362E-07
      I                 I
     ---               ---
C1=1.364E-10   C2=1.969E-10
     ---               ---
      I                 I
      I                 I
O - - - - - - - - - - - - - - - - O

READY
#
```

Example B. RUN of pi-network design program.

tells you that your desired loss is L times "this is greater than the minimum loss of XDB." Thus, you must re-enter data. Step 1360 sends you back to step 1060 which prompts for a new input impedance, output impedance, and desired loss. In step 1095, if the sign of R2 is not equal to -1, then you go to step 1100 and print out pictorial diagram in steps 1120 through 1180, with the values being printed in steps 1182 through 1186; minimum loss printout in step 1190; prompt for carriage return in step 1191, and in step 1195, you return to the calling routine.

Applications

Example A contains sample runs of the Electronics Calculator Routines. In the first run we have a resistive divider network (see Fig. 1) which has no R1 value, an E out of 3V with 12

```
READY
#

#
#RUN



   **************
ATTENUATOR DESIGN PGM

CHOOSE:

       1.TEE NET ATTEN
       2. PI NET ATTEN
? 1

   *** TEE NET ATTEN ***
INPUT IMPEDANCE? 1000
OUTPUT IMPEDANCE? 500
DESIRED LOSS -DB? 5

DESIRED LOSS 3.162277 TIMESGREATER THAN MIN LOSS
7.65551171 DB THUS YOU MUST REENTER DATA

INPUT IMPEDANCE? 1000
OUTPUT IMPEDANCE? 500
DESIRED LOSS -DB 10

O - - - R1 - - I - - R2 - - - O
               I
               I
              R3
               I
               I
O - - - - - - - I - - - - - - - O
       R1=725.31822
       R2=114.20711
       R3=496.904

MINIMUM LOSS=7.65551171  DB.
NEXT HIT C/R?

       **************
ATTENUATOR DESIGN PGM

CHOOSE:

       1.TEE NET ATTEN
       2. PI NET ATTEN
? 2

       *** PI NET ATTEN ***
INPUT IMPEDANCE? 1000


       *** PI NET ATTEN ***
INPUT IMPEDANCE? 1000
OUTPUT IMPEDANCE? 6E2
LOSS IN DB? 10

O - - I - - - R3 - - - I - - O
      I               I
      R1              R2
      I               I
O - - I - - - - - - - - - I - - O

R1=3174.56767
R2=885.098065
R3=1102.27005

MIN LOSS  = 6.4753125
NOW HIT C/R?
READY
#
```

Example C. Attenuator design routines.

```
#RUN

VOLTAGE TO DBM PROGRAM . . . . . WANT
INSTRUCTIONS? YES

   THIS PROGAM SUPPLIES THE UNKNOWN QUANTITYWHEN ANY TWO VALUES ARE GIVEN.

  ENTER KNOWN VALUES WITH THE VALUE AND THE UNKNOWNS AS 0 OR 1
THE PROGRAM WILL PROMPT FOR SOLUTION OF THE UNKNOWN
FINISHED? ENTER C/R?

IMPEDANCE IN OHMS? 50000
VOLTAGE IN V RMS? 0
LEVEL IN DBM ? -52

SOLVE FOR (DBM, VOLTS, IMP)? VOLTS
SOLUTION:
. . . . . VOLTAGE =0.01776172 RMS

IMPEDANCE=50000
 DBM =       -52
ANOTHER (1=YES, 0=NO)? 1

IMPEDANCE IN OHMS? 0
VOLTAGE IN V RMS? .775
LEVEL IN DBM  ? 0

SOLVE FOR (DBM, VOLTS, IMP)? IMP
SOLUTION  :
. . . . IMPEDANCE=600.625 OHMS

VOLTS =0.775
DBM = 0
ANOTHER (1=YES, 0=NO)? 1

IMPEDANCE IN OHMS? 500
VOLTAGE IN V RMS? .009
LEVEL IN DBM ? 1

SOLVE FOR (DBM, VOLTS, IMP)? DBM
SOLUTION:
. . . . . DBM= -37.9048496

VOLTS = 9.E-03
IMP = 500
ANOTHER (1=YES, 0 = NO)? 0

READY
#
```

Example D. DBM conversions.

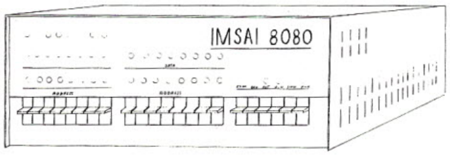
applied, and R2 - 1k Ohm. Next we read 3.67 volts peak-to-peak on an oscilloscope, and convert to its RMS value so we know what we'll read on a voltmeter. Next, we have a 115 volt reading on an ac voltmeter and solve for the correct value on an oscilloscope. Still calling from the Executive Routine we want to know the capacitive reactance of a .001 uF capacitor at 10 kHz. Next we need to fabricate a coil using a 1/4 form and #18 wire for 10 uH. In the next run we desire to find the capacitance of two conductors on opposite sides of a PC board. The conductors are 5 mils wide and 10 cm long. Finally, we need to find the resistance of 123 feet of #28 AWG copper wire. Last, we enter 0 and leave the Executive Routine.

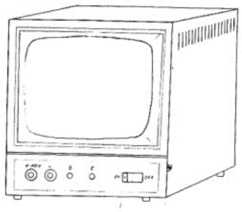
The pi-net run (Example B) shows the results of designing a 28 MHz impedance matching network for 125 Ohm output impedance from a transistor amplifier stage. The stage must match an impedance of 50 Ohms. The bandwidth for the network is to be around 9 MHz (Q of 3). Note the simulated schematic diagram printout.

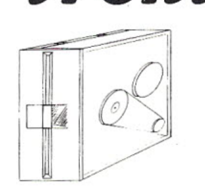
The next runs are concerned with attenuator design (Example C). First, the tee attenuator is desired to have a loss of 5 dB, but that is greater than the minimum loss, so the user is prompted to reenter. Since his impedances can't change, he is forced to use a different attenutation. The program reads out a simulated schematic calling out the resistors. Then it lists the resistance values required. Next pi-net attenuator design is accomplished.

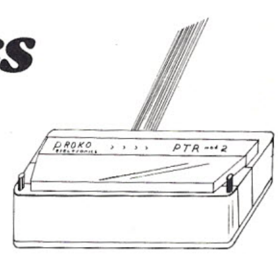
Example D is a sample run of the DBM program. First, we find the output voltage of a microphone that is spec'd at -52 DBM into 50k Ohms. Next we find the voltage level of a O DBM signal into 600 Ohms. And last, we find the DBM output of a 9 mV signal into 500 Ohms. ■

Quality by Design — *proko tronics*









IMSAI, Z-80 Kit

We can now offer the IMSAI micro computer with Z-80 power. By special agreement we are supplying the S.D. Sales Z-80 CPU card instead of the standard 8080 processor. For a really powerful system, combine this with the North Star disk system. Just power up and GO BASIC.

22 slot $745
6 slot $695

Quality Video Monitors

Looking for something to display 64 or 80 characters from a video board? We offer two, high bandwidth, 9" monitors with composite video inputs, in steel cases.

Sanyo VM-4209
$195
Hitachi VM-909
$195

Write for other sizes.

North Star Floppy Disc System

Accessible mass storage is a must for a viable computer system and North Star does it right. Their controller card contains all the ROM needed to get the system up and running. Just turn on the power, examine E900, run, and type GO BASIC. You can be up and running BASIC 8 seconds after power on. Disc drive & cables, controller kit BASIC & DOS.

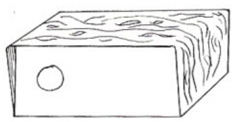
$680

PTR Mod 2 Paper Tape Reader

We keep improving our optical reader. It now comes with an acknowledge line to make interfacing simpler. Just set the PTR-2 under a study lamp, run the loader program, and pull thru a paper tape. Don't miss all that good software just because you don't have a reader.

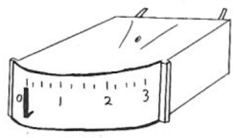
Assembled **$68** Kit **$55**

Dealer inquiries invited.



Nice walnut grained aluminum boxes about 4 x 4 x 2". Some holes, but good for power supplies and other small projects.

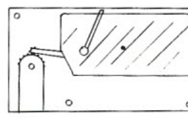
$1.25 each
5/$5.00



3 MA full scale meters — use with a 10K resistor for 0 to 30V DC. A good universal meter for current or voltage.

$3.50 ea.
2/$6.00

USED MUFFIN FANS
$9.00 ea.



Mechanical paper tape read-head, 8 level star-wheel type. A good place to start for a homebrew paper tape reader for standard 1" tape.

Like new $9.50 ea.

proko tronics

439 Marsh Street
San Luis Obispo, California 93401

Check or money order only. If you are not a regular customer and your order is large please send either a cashier's check or a postal money order, otherwise there will be a delay of two weeks for the check to clear. All items post paid in the U.S. Calif. residents add 6% tax. Prices subject to change without notice. $10 minimum order.

---

cybercom BOARDS

MB-1 MK-8 Computer RAM, (not S-100), 4KX8, uses 2102 type RAMs, PCBD only $22

MB-3 1702A EROM Board, 4KX8, S-100, switchable address and wait cycles, kit less PROMS $65

MB-4 Basic 4KX8 ram, uses 2102 type rams, may be expanded to 8KX8 with piggybacking, S-100 buss. PC board $30

MB-6 Basic 8KX8 ram uses 2102 type rams, memory protect in 256 to 8K switchable S-100 buss. PCBD $35

MB-8 2708 EROM board, S-100, 8KX8 or 16KX8 kit without PROMS $85

IO-2 S-100, 8 bit parallel I/Oport, ⅔ of board is for kludging. Kit...... $55 PCBD $30

VB-1 64X 16 video board, upper lower case Greek, composite and parallel video with software, S-100. Kit...... $189.00 PCBD $35

SP-1 Music synthesizer board, S-100, computer controller wave forms, 9 octaves, 1V rms ½% distortion, includes software kit $250

Altair Compatible Mother Board, 11 x 11½ x ⅛". Board only ... $45 With 15 connectors $105

Extender Board full size. Board only $9
With connector $13.50

Solid state music Cybercom boards are high quality glass board with gold finger contacts. All boards are check for shorts. Kits only have solder mask. 90 day guarantee on Cybercom kits.

Waneco Boards

MM-1 8KX8 fully buffered, S-100, uses 2102 type rams. PCBD $30

Mother Board 12 slot, terminated, S-100, board only $40

Special 8080A processor & support chip set kit includes one of each; all parts prime 9080A (8080A); 8228; 8212; 8216; 8251; 8214; 8224; 8225. Special $75.00

Special 2101AL-4 1K x 1 ram ⅓ less power than 21L02 type rams, with power down, prime from NEC. Ea. 2.00; 32 ea. 1.80; 64 ea. 1.70; 128 ea. 1.60; 256 ea. 1.50; 512 ea. 1.45.

9080A AMD 8080A (Prime)	25.00
8212/74S412 Prime	4.00
8214 Prime	8.30
8216 Prime	4.95
8224 Prime	5.00
8228 Prime	8.90
8251 Prime	14.50
8255 Prime	14.50
1702A-6 AMD 4702A Prime	8.00
TMS-6011 UART Prime	6.95
2513 Char Gen Upper Prime	11.00
2513 Char Gen Lower Prime	11.00
1702A Intel Not Prime	8.00

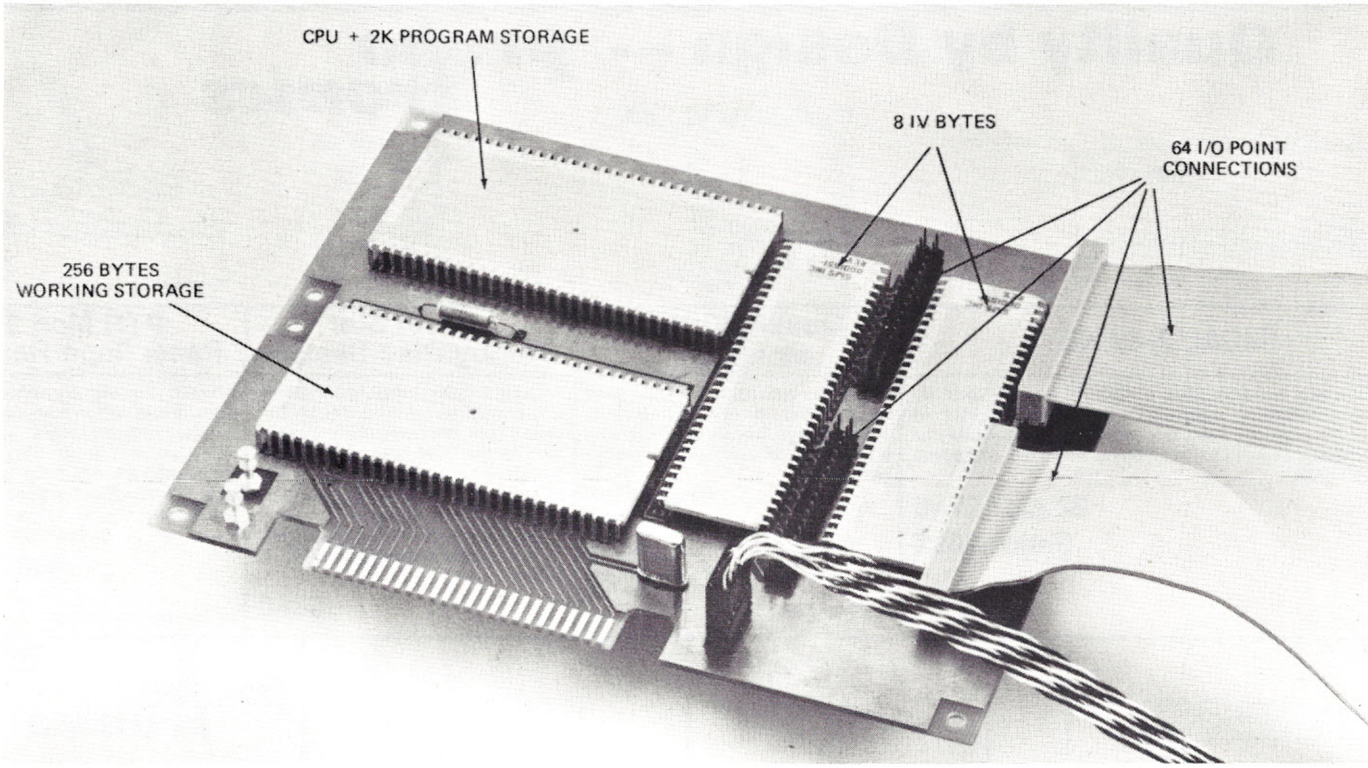
8T10	2.00	8T97	2.00	80L97	1.50
8T13	2.50	8T110	2.00	81L22	1.50
8T16	2.00	5309	8.00	82L23	1.90
8T20	2.00	5312	4.00	85L51	2.50
8T24	2.50	5313	4.00	85L52	2.50
8T26	2.75	5320	5.95	85L63	1.25
8T34	2.50	5554	1.90	86L70	1.50
8T37	2.50	5556	2.50	86L75	1.90
8T38	2.50	5055	1.60	86L99	3.50
8T80	2.50	MC4044	2.25	88L12	.80

74L00	.25	74LS00	.40	1101	1.25
74L01	.25	74LS01	.50	1103	1.25
74L02	.25	74LS02	.40	2101	4.50
74L03	.25	74LS03	.40	2111-1	3.75
74L04	.30	74LS04	.45	2112	4.50
74L05	.40	74LS05	.45	2602	1.60
74L06	.30	74LS08	.40	4002-1	7.50
74L08	.40	74LS10	.40	4002-2	7.50
74L09	.40	74LS12	.55	MM5262	1.00
74L10	.30	74LS20	.40	7489	2.00
74L20	.35	74LS22	.45	74200	4.95
74L26	.40	74LS27	.45	74C89	3.00
74L30	.40	74LS30	.40	82S06	2.00
74L32	.45	74LS37	.60	82S07	2.00
74L42	1.50	74LS38	.60	82S17	2.00
74L51	.35	74LS42	1.50	8223	2.50
74L54	.45	74LS51	.40	82S23	3.00
74L55	.35	74LS54	.45	82S123	3.00
74L71	.30	74LS55	.40	82S126	3.50
74L73	.55	74LS73	.65	82S129	3.50
74L74	.55	74LS74	.65	82S130	3.95
74L75	1.20	74LS76	.65	82S131	3.95
74L78	.90	74LS151	1.55	IM5600	2.50
74L85	1.40	74LS174	2.20	IM5610	2.50
74L86	.75	74LS175	1.95	IM5603	3.00
74L89	3.50	74LS192	2.85	IM5604	3.50
74L90	1.50	2501B	1.25	IM5623	3.00
74L91	1.50	2502B	3.00	IM5624	3.50
74L93	1.70	2507V	1.25	MMI6330	2.50
74L95	1.70	2510A	2.00	DM8573	4.50
74L98	2.80	2517V	1.25	DM8574	5.50
74L123	1.50	2519B	2.80	DM8575	4.50
74L164	2.50	2532B	2.80	DM8576	4.50
74L165	2.50	2533V	2.80	DM8577	3.50
74L192	1.25	DM8131	2.50	DM8578	4.00
74L193	1.20	N8263	3.50	2.4576 MHZ	
MH0026	2.95	MC1489	1.50	XTAL	7.20
MC1488	1.50	DM8837	1.50		

MIKOS

419 Portofino Drive
San Carlos, California 94070
Please send for IC, Xistor and Computer parts list.

Check or money order only. If you are not a regular customer and your order is large please send either a cashier's check or a postal money order, otherwise there will be a delay of two weeks for the check to clear. All items post paid in the U.S. Calif. residents add 6% tax. Money back 30 day guarantee. We cannot accept returned IC's that have been soldered to. Prices subject to change without notice. $10 minimum order.



The Scientific Micro Systems MicroController, a powerful computer for control applications. It has only 8 distinct instructions. Courtesy of Scientific Micro Systems, Mountain View, CA.

Dr. Lance A. Leventhal
Engineering and Technology Department
Grossmont College
El Cajon CA 92020

Understand Your Computer's Language

... Part 2: A Further Look at Instruction Sets

The instruction set of a computer determines what that computer can do in a single instruction cycle. Sophisticated instructions will allow a computer to perform a variety of tasks without further direction. Operations that are not implemented as single instructions must be performed by a series of instructions. Such a series requires extra time to fetch and decode the instructions and extra work on the part of the programmer. The instruction set therefore has considerable effect on computer throughput and the ease with which the computer can be programmed. This article will describe the types of operations that an instruction set must perform and how small computers implement those operations. We will then establish some criteria for comparing the instruction sets of different computers.

Categorizing Instructions

Categorizing computer operations is a difficult matter since the architecture and applications of com-

puters vary so widely. We may roughly divide the fundamental tasks into four categories:

a. Data manipulation operations which actually change the data in some way.

b. Data transfer operations which move data from one place to another without changing it.

c. Program manipulation operations which alter the sequential flow of program control.

d. Status management operations which perform status or overhead functions.

Although some computer instructions result in only one type of operation, others may result in several types of operations in a single cycle. We will first discuss instructions which perform one type of operation and will then briefly consider the extension to instructions that combine types. Note that only data manipulation operations actually change the data; the other operations are "paper shufflers" which make sure that the computer performs the correct data manipulation operations in the proper order.

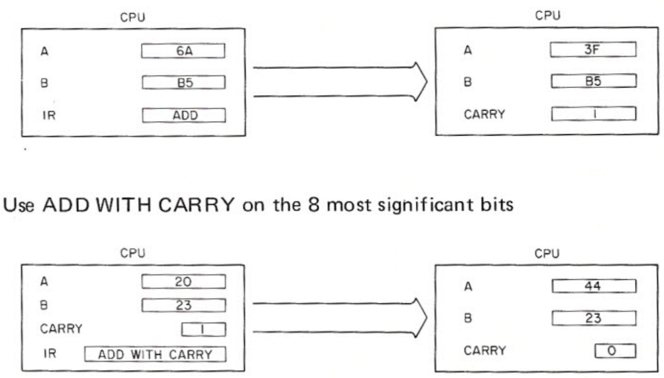
Data Manipulation Operations

Data manipulation operations form the heart of most programs since these operations actually do the processing. We may subdivide the instructions in this category as follows:

- Arithmetic instructions
- Logical instructions
- Shift instructions
- Comparison instructions
- Special-purpose instructions

Arithmetic instructions are the simplest to understand. ADD and SUBTRACT have obvious functions. Note that SUBTRACT can also be used to determine if two quantities are equal; we subtract one from the other — if the result is zero, the two were equal. MULTIPLY and DIVIDE are not usually available as single instructions on small computers (the Texas Instruments 9900 is one of the few microprocessors that has these instructions); we must implement these operations as entire series of ADD or SUBTRACT instructions much as we would do them by hand. Other instructions which are usually present are ADD WITH CARRY and SUBTRACT WITH BORROW; these allow us to perform multiple-word arithmetic by using the CARRY or BORROW to transfer information between words (see Fig. 1). INCREMENT and DECREMENT add or subtract 1; we use these instructions to increment or decrement counters, indexes, and indirect addresses. INCREMENT and DECREMENT are not only shorter and faster than the regular ADD and SUBTRACT, but they also do not affect the CARRY so that they can be used in loops which perform multiple-word arithmetic. DECIMAL ADD is convenient in applications like calculators, games, and cash

Problem: Add 206A and 23B5 (hexadecimal) using an 8-bit computer
Solution: Use ADD on the 8 least significant bits

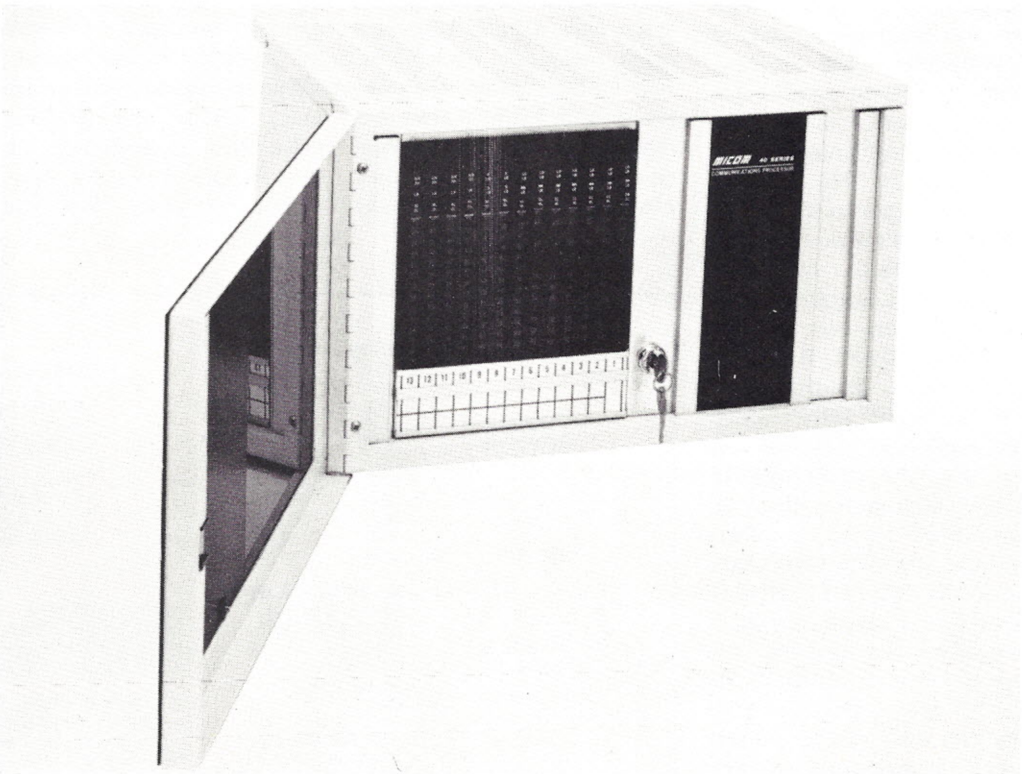


Obviously we can continue this procedure using ADD WITH CARRY for longer numbers.

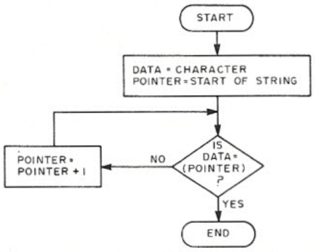
Fig. 1. Using the ADD and ADD with CARRY instructions.

registers where all the data is decimal rather than binary.

Logical instructions may not seem as immediately useful as arithmetic instructions but are an essential part of most computer applications. The most commonly used instruction is AND; OR, EXCLUSIVE OR, and COMPLEMENT are also usually available. Logical AND is used to mask bits, i.e., to remove one or more bits from a word. For example, assume that we want to see if a switch attached to data line #3 is closed (0) or open (1). The procedure is to fetch the switch data from the input port and AND it with a mask which has a 1 in bit position 3 and zeroes elsewhere. Since anything ANDed with zero is zero, the result depends only on the status of the one switch; the result is zero if and only if that switch is closed. We may also use logical AND to clear bits (e.g., to turn a positive display off) by ANDing the



The MICOM Communications Microcomputer, a system specifically designed for communications applications. Courtesy of Micom Systems Inc., Chatsworth, CA.



Remember that POINTER is an address and (POINTER) is the contents of that address.

Fig. 2. Flowchart of comparison procedure.

data with a mask which has zeroes in the positions to be cleared and ones elsewhere. The other ones leave those positions in the same states as before (try it). Logical OR can be used to combine fields and to set bits (i.e., by ORing with a 1 bit); logical EXCLUSIVE OR can reverse bits and form checksums or parity; COMPLEMENT is necessary for subtraction and for handling peripherals (such as common-anode displays) which send or receive data with negative logic (i.e., 0 volts is on or a 1, +5 volts is off or a 0).

Shift instructions allow us to place bits or groups of bits where they can easily be handled. Shift instructions are also an essential part of multiplication and division, scaling, serial to parallel and parallel to serial conversions, and many mathematical functions. LOGICAL SHIFT moves the data to the left or right and fills the empty bit with a zero. LOGICAL SHIFT LEFT is equivalent to multiplying by two (try it); LOGICAL SHIFT RIGHT is equivalent to dividing by 2. ARITHMETIC SHIFT shifts the data but preserves the sign bit. The ARITHMETIC SHIFT instruction allows us to shift signed numbers without changing their signs; an ARITHMETIC RIGHT SHIFT of a signed number is equivalent to dividing the number by 2. CIRCULAR SHIFT moves the data as if the two ends of the number were connected, either directly or through the CARRY.

Comparison instructions allow us to compare numbers in various ways by setting the flags without actually changing any data. The data is then preserved for further comparisons or other operations. Typical instructions will perform a subtraction or a logical AND but will not store the result anywhere. We may then look for a particular character or pattern (such as a carriage return) in a string of data without reloading registers prior to each subtraction or logical AND.

The procedure for a single character is (see flowchart in Fig. 2) as follows:

Step 1. place the required character in the accumulator.
Step 2. compare to element in string.
Step 3. if they are equal, the character has been found. Otherwise get next element from string and repeat step 2.

The same basic procedure can handle longer comparisons. The program checks for equality with a JUMP ON ZERO instruction.

Special-purpose instructions are useful for particular applications. Such instructions may include keyboard scans, decimal corrections, text editing, communications error-checking, and bit manipulations. These instructions will not be used very often but may be extremely convenient in some situations.

Data Transfer Operations

We can further divide the instructions which implement data transfer operations as follows:

- Memory transfer instructions
- Input/output instructions
- Register transfer instructions
- Stack instructions

Memory transfer instructions move data to or from the memory. LOAD transfers data from the memory to a register; STORE transfers data from a register to the memory. Note that neither LOAD or STORE changes the source of the data — only the destination. CLEAR is a special transfer operation which places zero in the destination.

Input/output instructions transfer data to or from I/O devices. IN or READ is an input operation, OUT or WRITE an output operation. Processors like the Motorola 6800 and MOS Technology 6502 which do not distinguish I/O devices from memory locations will have no special I/O instructions; memory transfer instructions (e.g., LOAD and STORE) with the appropriate addresses will serve the same purposes. Note that we seldom want to transfer one character at a time. Peripherals like card readers, line printers, CRT displays, paper tape readers, and floppy disks typically transfer entire blocks of data. Specialized block transfer I/O instructions are an important feature of most modern computers and are available on newer processors such as the Zilog Z-80.

Register transfer instructions transfer data from one register to another. The most common operations are loading an accumulator from a general-purpose register and storing the contents of an accumulator in a general-purpose register. The specific instructions have various names, including LOAD, STORE, MOVE, COPY, TRANSFER, TRANSMIT, and REPLACE. An EXCHANGE instruction actually exchanges two registers without destroying the contents of either one; such an instruction will replace several simple register transfer instructions since the contents of one register must be saved somewhere if two registers are to be exchanged properly.

Stack instructions transfer data to or from a section of memory and update a pointer so that the section appears to be a stack or last-in, first-out memory. PUSH adds data to the stack; POP or PULL

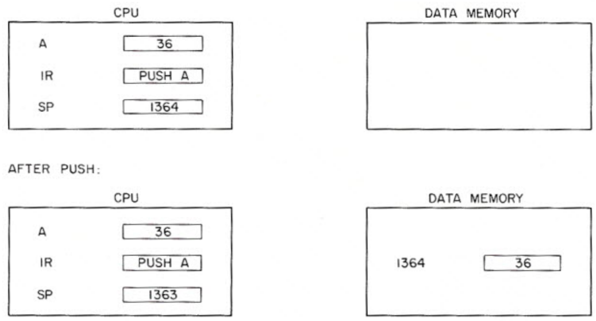
```
          INTEL 8080
          LDA     COUNT     ;    GET COUNTER
          MOV     B,A
          LXI     H,START   ;    POINTER = START OF ARRAY
NEWMX:    MOV     A,M       ;    GET NEW MAXIMUM
NEXTE:    DCR     B         ;    COUNTER = COUNTER - 1
          JZ      DONE
          INX     H         ;    POINTER = POINTER + 1
          CMP     M         ;    IS NEXT ELEMENT MAXIMUM?
          JC      NEWMX     ;    YES, REPLACE MAXIMUM
          JMP     NEXTE     ;    NO, KEEP LOOKING
DONE:     HLT

          MOTOROLA 6800
          LDAB COUNT        GET COUNTER
          LDX #START        POINTER = START OF ARRAY
NEWMX     LDAA X            GET NEW MAXIMUM
NEXTE     DECB              COUNTER = COUNTER - 1
          BEQ DONE
          INX               POINTER = POINTER +1
          CMP A X           IS NEXT ELEMENT MAXIMUM?
          BCS NEWMX         YES, REPLACE MAXIMUM
          BRA NEXTE         NO, KEEP LOOKING
DONE      SWI
```

Example 1.

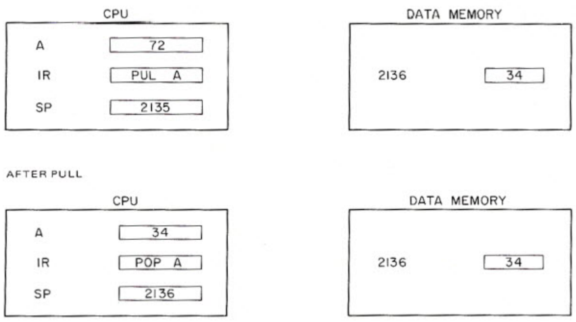
I. PUSH
A push results in
((SP)) = (Register)
(SP) = (SP)-1

BEFORE PUSH:

AFTER PUSH:



II. PULL (or POP)
A pull results in
(SP) = (SP) + 1
(Register) = ((SP))



Note that the data in 2136 does not change; however, the next PUSH operation will place new data in that address if the stack pointer has not been changed.

Fig. 3. The Pushdown Stack on the Motorola 6800.

removes data. Fig. 3 shows how the Motorola 6800 performs such transfers. These instructions differ from regular memory transfer operations in that they use the stack pointer as the memory address and update its value with each use. Stack instructions actually perform some arithmetic (increment or decrement) in addition to their data transfer functions.

Data transfer operations are the most common operations in almost all programs. Most of the instructions move data from one place to another — from a register or memory to the accumulator and back again, from memory to address or data registers. In Example 1, the programs find the maximum of an array of unsigned 8-bit numbers. Note how many instructions simply move data around, adjust counters or pointers, or select alternate program sequences. The actual processing is a small part of the program.

Program Manipulation Operations

We can divide the instructions which perform program manipulation operations as follows:

a. Unconditional jump instructions which always place a new value in the program counter.

b. Conditional jump instructions which will only place a new value in the program counter if the specified conditions are met.

c. Subroutining instructions which allow a program to transfer control to and from subroutines.

d. Halts and no operations which have no effect on anything except the sequencing of instructions.

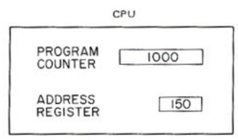
Some computers treat the program counter just like any other register. Then register and memory transfer instructions can replace explicit program manipulation instructions. Such replacements, however, can make a program almost impossible to debug or understand, since the programmer must distinguish between transfers of control and ordinary transfers of data.

Conditional and unconditional jump instructions are really transfer operations where the destination register is the program counter. Unconditional jumps are useful for reaching programs from fixed reset or interrupt service addresses, returning to a starting point, or allowing an external user to control the starting point (e.g., as in the GO command in many simple monitors). Conditional jumps form the decision-making capability of the computer since they allow it to choose among sequences of instructions depending on internal conditions or external inputs. The most common conditional jump instructions are JUMP ON ZERO and JUMP ON NOT ZERO, which we use to control loops and search for particular data items, and JUMP ON CARRY and JUMP ON NOT CARRY, which we use to compare values and to examine single bits of data. Fig. 4 shows the results of a conditional jump instruction.

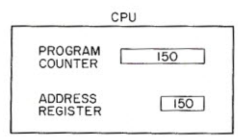
Subroutining instructions differ from jumps only in that they provide a way back to the original program sequence. The main program then can transfer control to a subroutine which will return control to the main program upon its completion. The instruction which transfers control to the subroutine must save the current value of the program counter so that the subroutine can find that value and use it as a return address. Different CPUs use different methods to save the return address; they may place it in program memory (JUMP AND MARK PLACE), a register (JUMP AND LINK), or the pushdown stack (CALL or JUMP TO SUBROUTINE). The pushdown stack (see Fig. 3) will require some read/write memory if it is located in external RAM (as in the Intel 8080 or Motorola 6800) but has the advantage that it can be used repeatedly or even recursively without causing any problems. Some CPUs have other subroutining instructions such as RETURN, which restores the old value of the program counter, TRAP or SOFTWARE INTERRUPT, which forces a jump to a specified location (used to indicate hardware errors or to debug programs), and RETURN FROM TRAP or RETURN FROM INTERRUPT, which reverse the trap or interrupt process.

HALT and NO OPERATION are surprisingly useful instructions. HALT stops the program counter from incrementing and allows the CPU to wait for an external signal. NO OPERATION does

Problem: Execute the instruction JUMP ON CARRY 150 (after instruction fetch).

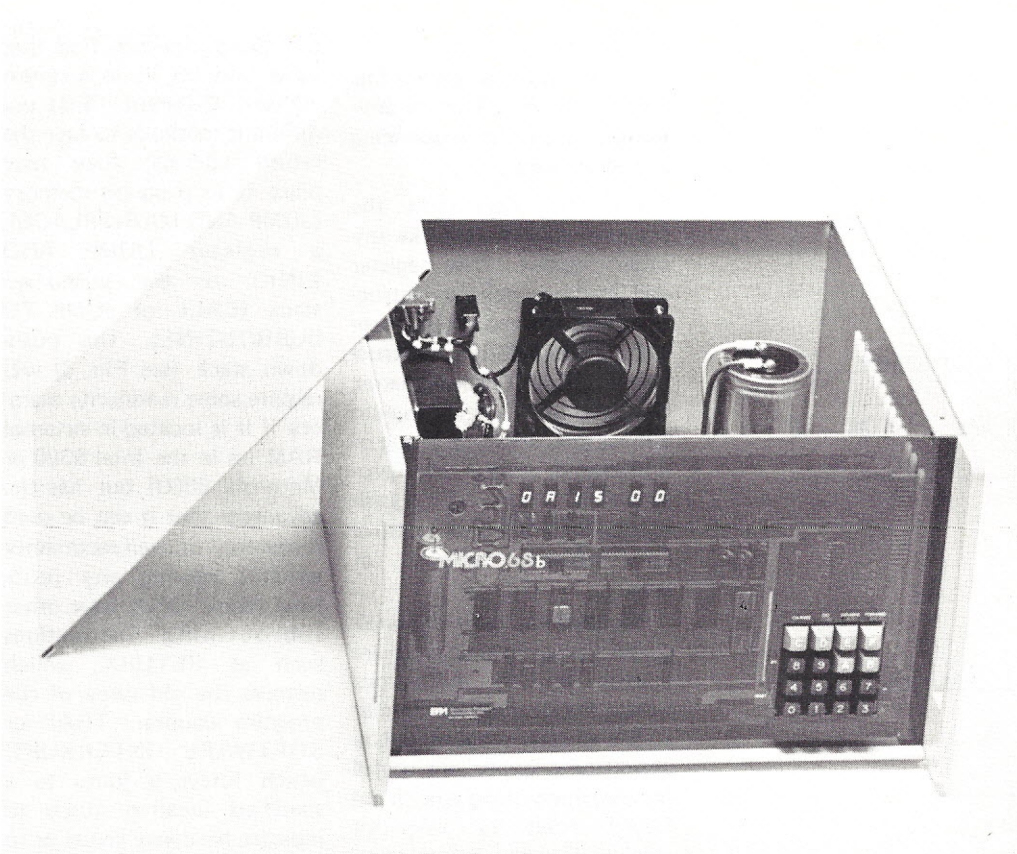


If CARRY = 1, (PROGRAM COUNTER) = (ADDRESS REGISTER) so CPU will fetch next instruction from memory location 150



If CARRY = 0, program counter is unchanged so CPU will fetch next instruction from memory location 1000

Fig. 4. The Conditional Jump Instruction.



The Electronic Product Associates Micro 68 b, a microcomputer based on the Motorola 6800 microprocessor. Courtesy of Electronic Product Associates, San Diego, CA.

nothing except increment the program counter; it can provide a delay, equalize the execution time of alternate instruction sequences, replace erroneous instructions, or leave space for corrections or additions.

Program manipulation operations, like data transfer operations, do not really process any data. All they do is tell the computer which processing instructions to execute. Note the amount of overhead involved in the operation of the computer. More cycles are spent moving data and changing the program counter than are spent in manipulating data. This inefficiency is part of the price we pay for the flexibility of the computer; specialized hardware can perform a task faster but is much more difficult to change, correct, or extend.

Status Management Operations

Status management operations include a variety of instructions which change the status without affecting any data or changing the program sequence. Typical tasks include allowing or disallowing (enabling or disabling) interrupts, selecting pages of memory or modes of operation, designating register functions, protecting or unprotecting the memory, and returning control to a supervisor program. These operations form a very small part of user programs.

Combined Instructions

As we mentioned, there is no reason why a single instruction cannot perform several tasks. One instruction could, for example, perform an addition, a shift, a masking operation, and a conditional jump. The problem is to find combinations that the programmer can use in a variety of situations. Some useful ones are block input/output, a loop control instruction which decrements a counter and causes a jump if the result is not zero (i.e., DECREMENT AND JUMP ON NOT ZERO), and an add and shift instruction which can form one step of a software multiplication. Unfortunately, most combined instructions are rather difficult to use; programmers will need them so seldom that they may forget the precise sequences involved. Combined instructions also make programs difficult to follow, debug, and document.

Microprocessor Instruction Sets

Microprocessors typically have small and simple instruction sets by comparison to today's minicomputers and large computers. Microprocessors usually have 40 to 80 different instructions; most lack specific instructions for multiplication and division, floating point arithmetic, multiple-word operations, bit manipulations, complex comparisons and conditional jumps, and block transfer I/O. However, today's microprocessors do have more sophisticated instruction sets than older minicomputers such as the DEC PDP-8 and IBM 1130. New devices like the Texas Instruments 9900 and Zilog Z-80 have some of the instructions that make the newer minicomputers so powerful. Improved instruction sets will in the future surely make microcomputers much easier to program and far more versatile.

However, a large instruction is not necessarily the answer. Some applications may be easier to implement when an instruction set only has a few very powerful instructions. The Scientific Micro Systems Microcontroller (see photo) works well in floppy disk controllers and switching systems, yet it has only 8 instructions. It could not act as a general-purpose computer but it can outperform far more expensive computers in specific applications.

Computers with specialized instruction sets may be useful in some types of applications. The MICOM communications microcomputer (see photo) will perform such functions as routing messages, combining channels, converting codes, speeds, and protocols, and providing voice response. These tasks would be very difficult to program on a more general computer. Future computers at all levels may well contain many special-purpose processors with instruction sets specifically designed for particular tasks.

Comparing Instruction Sets

The newcomer to the computer field often faces many conflicting claims. Is a system like the Electronic Product Associates Micro 68b (see photo) the best because of the Motorola 6800's instruction set? Or is a Mits Altair 8800b (see photo) better because of the Intel 8080 instruction set? Is a Zilog Z-80 based-system best of all because the Z-80 has more of everything?

Almost all computer manufacturers claim that their machine's instruction set is not only the most powerful but also the easiest to learn and use. We will examine a few of the criteria for evaluating instruction sets and will show how some microprocessors stand up to those criteria. Instructions sets are complex and difficult to measure; we will only mention a few of the considerations involved.

An instruction set is powerful if relatively few instructions are necessary to perform common tasks. The important factors in determining the power of an instruction set are:

(1) How many separate instructions are there?

(2) How complete is the set?

(3) Are there single instructions for performing such common operations as loop control, table accesses, and arithmetic functions?

(4) What restrictions are there on the instructions?

(5) Are there special-purpose instructions for a particular application?

All of these questions are hard to define and closely interrelated.

Number of Instructions

The sheer number of instructions is a relatively simple measure of the power of an instruction set. Clearly, if a computer has more instructions, we are more likely to find an instruction suited to a particular task. Programs will generally be shorter, they will occupy less memory, execute faster, and be easier and quicker to write. The fetching of instructions is purely an overhead function for the computer. This overhead is the price for the flexibility of the computer but short, powerful instructions keep the overhead from seriously affecting throughput.

Considerations of the number of instructions favor the Intel 8080 and (even more) the Zilog Z-80. The Z-80, in particular, has far more instructions (158) than any of the other processors. The remaining question, of course, is how much use the programmer will get from the extra instructions. Are the additions important ones, like memory and percentage keys on a four-function calculator or are they just gadgets or trim like the wheel covers and racing stripes that the automobile manufacturers advertise so heavily?

One problem in using the criterion of number of instructions is that manufacturers seldom agree on what constitutes a distinct instruction. Do, for example, instructions which provide the same operation with different addressing methods count separately? Intel separates them (i.e., ADD REGISTER, ADD IMMEDIATE) when describing the Intel 8080 instruction set while Motorola does not. What about instructions which perform several operations in one cycle — e.g., a decrement, a conditional jump, and perhaps a shift? The number of instructions may be difficult to determine and instruction sets may not be directly comparable.

Furthermore, some instructions may be used so seldom that they really are not very valuable. For example, few Intel 8080 programmers will have the occasion to use the conditional jumps which depend on the parity bit while few Motorola 6800 programmers will use the ones which depend on the overflow bit. Some instructions are so rare that the user would never notice their omission while others are so common that they appear in virtually every sequence. Evaluations of instructions sets must consider not only how many instructions the computer has, but also how useful they are.

Look at some of your own programs in assembly language. A few common instructions will make up 90% or more of the programs. On the other hand many instructions may not appear at all. Adding more instructions to the list of those that never get used anyway does not make programming simpler.

Completeness

The completeness of an instruction set is also difficult to measure; the user will only be immediately aware of a gaping hole. A processor which lacks specific instructions for common arithmetic functions, logical operations, shifts, conditional or unconditional branches, or subroutining is clearly less powerful than one that has those instructions. But most modern CPUs have all these features and the completeness of their instruction sets depends on less obvious considerations. Common areas in which there are differences include shift and conditional jump instructions. An instruc-

The MITS Altair 8800b, a microcomputer based on the Intel 8080 microprocessor. Courtesy of Mits Inc., Albuquerque, NM.

tion set that has all the useful operations will be more powerful than one that does not. For example, the Intel 8080 lacks specific instructions for arithmetic and logical shifts and for conditional branches which depend on combinations of flag bits; the Motorola 6800, on the other hand, has specific instructions for these operations.

The lack of specific instructions for an operation does not mean that the computer cannot perform that operation. Rather it means that the operation will require several instructions. This slows down the computer since it must spend more time in the overhead function of fetching instructions. Multiple instruction sequences also occupy more memory and are harder to write and debug. Compare, for example, the implementation of a four-bit shift on a processor that has a multiple-bit shift instruction (like the National Pace) and one that has only single-bit shifts (like the Intel 8080 or Motorola 6800). The latter devices must fetch, decode, and execute four instructions to perform the same operation that the National Pace can do with one.

Some combined operations are so common that a specific instruction can save a great deal of time and memory. Examples include loop control (decrementing a counter and branching), checking for a character (subtracting the character and branching), or accessing a table (calculating the address and fetching the data). Few small computers have single instructions to handle these tasks but the next generation of microprocessors will probably have such features. Table accesses are particularly awkward on small computers — the Intel 8080 at least has an address-length addition (DAD) for handling this problem; the Motorola 6800's indexing is useless for this task since the 16-bit starting address of the table (the fixed part of the operation) cannot fit into the 8-bit offset.

Many computers have restrictions on their instruction sets which greatly limit their power and flexibility. Some common restrictions are: (a) Only allowing operations between registers. Separate instructions are then necessary to load, update, and store the contents of the various registers that the instruction uses. (b) Restricting operations to handling data in the accumulator. Again extra instructions are necessary to load and store the contents of the accumulator. (c) Limiting the addressing range. Some instructions may only use addresses that are on page zero or the current page. Addresses that are further away can only be reached through an extra stage of addressing.

Almost all computers have some of these restrictions. The Intel 8080, for example, only permits arithmetic and logical operations between the accumulator and either a register or fixed data in ROM. Furthermore, shifts and other instructions can only use data in the accumulator. The Motorola 6800 allows many operations which directly handle data in memory, but restricts indexed offsets to 8 bits and conditional branches to nearby locations that can be reached via relative addressing. Of course, one reason why the Motorola 6800 allows operations on data in memory is that it has no general-purpose registers.

Some common applications are easier to implement if the instruction set has specific instructions. Examples are decimal arithmetic for calculators, parity and other error-checking features for communications, text analysis for editing, and bit manipulation for control applications. Special minicomputers for particular applications are available and special microprocessors will probably be available in the next few years. Multiprocessor systems may well consist of a network of specialized CPUs.

Consistency and Straightforwardness

The power of the instruction set is not the only important measure for comparisons. Another question is how easy is the instruction set to use. Clearly power is an important factor here since a more powerful instruction set means that the programmer must write and debug fewer instructions. But there are other factors to consider such as consistency, simplicity, and straightforwardness, which can make the programmer's tasks much easier.

Consistency is one key to producing an instruction set that is easy to use. Each special case that treats one register differently from the others, handles one condition or transfer in a distinct way, or only permits certain combinations, makes a computer harder to program. The programmer must remember each of these limitations and allocate resources so as to handle them. Careful ordering of operations becomes vital and the debugging of programs becomes slow and cumbersome. Inconsistency is perhaps the greatest weakness of the Intel 8080 and Zilog Z-80. Both have a large number of special instructions which affect particular registers in a unique manner. The programmer must be sure to assign data and addresses to the proper registers in the correct order. Such inconsistencies make an instruction set difficult to learn and to use. The Motorola 6800, on the other hand, has a relatively consistent instruction set.

Simplicity and straightforwardness really mean that the instruction set should perform operations in a way that the human programmer can easily understand. The computer itself has no preferences, never gets confused, and never fails to make the proper distinctions. Programmers, on the other hand, prefer actions to be simple and well defined, distinct operations to be clearly separated, and addresses and data to be easily distinguished. Processors like the Intel 8080 and (far more) the RCA Cosmac, which rely heavily on indirect addressing and other methods which do not explicitly identify data and addresses are difficult for programmers to learn and use. Too often in the past, the manufacturer has expected the programmer to adapt to an obscure and inconsistent instruction set. Hopefully, future computer instruction sets will be designed with a proper consideration of human factors. Such factors are particularly important when programs are written in assembly or machine language.

A brief review of instruction sets cannot serve to fully catalog them. The instruction set is clearly one determining factor in the power and usefulness of a computer. An evaluation of instruction sets must consider more than just the number of instructions. It must also consider the instruction set from the point of view of completeness, usefulness, consistency, simplicity, and suitability for particular applications. ■

References

Osborne, A., *An Introduction to Microcomputers, Volume 1: Basic Concepts*, Adam Osborne and Associates, Berkeley, Ca., 1975.

Leventhal, L.A., "Put Microprocessor Software to Work," *Electronic Design,* August 2, 1976, pp. 58-64.

Cushman, R.H., "Microprocessor Instruction Sets," *EDN*, March 20, 1975, pp. 35-41.

Galletti, C., "Will the Z-80 Crush All Competitors?", *Kilobaud*, February 1977, pp. 50-52.

Godding, P., "Is the Z-80 the Wave of the Present?", *Kilobaud*, January 1977, pp. 20-24.

Allison, D.R., "A Design Philosophy for Microcomputer Architectures," *Computer*, February 1977, pp. 35-41.

Dollhoff, T., "Techniques for the Intel 8080 and Motorola 6800," *Digital Design*, November 1976, pp. 56-69.

Enter the Audible Computer!

. . . build this simple tone generator interface

John E. Stith
4486 Escarpardo Way
Colorado Springs CO 80917

If you have an unused bit on an I/O port in your system, you may want to try this under-ten-dollar project. An audible indicator has a lot of applications and I'm sure you will discover many more.

I decided to investigate the possibilities of such a device because I spend most of my computer time at a cheap keyboard and a TV typewriter. The keys are extremely inconsistant. Several characters cannot be entered without using a hammer, and a few others stutter. Several characters are entered with only one key depression, and I don't have a repeat key. The problem was compounded by the fact that I had to constantly watch the CRT to see if what I thought I typed was what was actually entered.

The solution would be an audible indication after every character that was input to the computer. This is not a new idea. Keypunches, cash registers, and other devices use the same technique.

The uses I've considered include the following: 1.) Acknowledge every character input from the keyboard and/or indicate you're nearing the end of a line (such as a typewriter bell). 2.) Have it used by your character output program; it could sound the alarm whenever an ASCII bell control character was found in the output. (Now you don't have to buy a $750 teletypewriter for the $10 bell.) 3.) Hams could use it as an international code practice aid. (Or make their ten-minute I.D. even more noticeable.) 4.) Signal that a long job has just finished. 5.) Signal that an end-of-file was just reached from your cassette or floppy disk. 6.) Use it with computer-assisted instruction to announce correct or incorrect answers, or ask questions like, "How many notes did you just count?" 7.) Monitor the progress of a long job by using a bell in each loop.

No doubt you or your children could come up with dozens of uses in the context of computer (bounce, bounce, bounce) games.

Building It

Now, on to the construction details. The three constraints I used were: a.) I wanted it cheap. b.) I wanted it convenient. (TTL-compatible, low-power, and no rf interference.) c.) The alarm had to be off if the computer was reset, without restarting the initialization program. (More about this later.)

The device I finally selected is the 5 volt version of the dip-alarm from Projects Unlimited, and needs only 20 mA. It is solid state, so it produces its 400 Hz tone without any rf interference, and fits in a standard 16-pin dip socket.

Since the current requirement is about double the typical TTL gate load, I used a TTL inverter driver 7416 IC between my output port and the alarm. The 7416 can sink 40 mA, and it only needs to do so while the alarm is on. The schematic and parts list are shown in Fig. 1.

Operating It

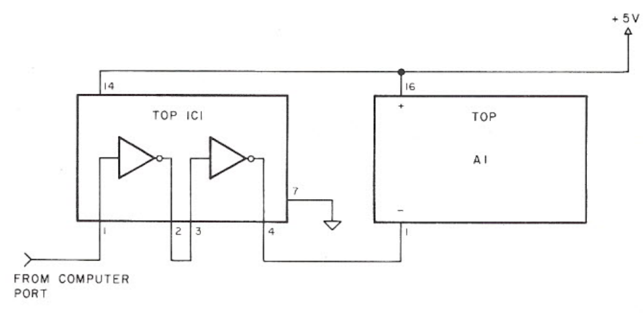
Control of the device is entirely by software. This design assumes that a one written to the output port turns the device off, and a zero turns it on. The reason this might seem logically reversed is my requirement that the device be off if I hit reset. My system uses the Motorola MC6820 Peripheral Interface Adapter (PIA). When reset, all the I/O lines are set to be input lines, until programmed by the initialization program. An input line represents almost no load to the 7416 input, so the 7416 input will be floating. Due to the nature of TTL, the floating input will be drawn toward five volts by the 7416 itself; so in effect, it sees a one as an input. Therefore, an input line or an output one represent off, and the output zero will turn the device on.

The alarm is connected so that the TTL device can sink the required current by holding its output to zero volts, but cannot supply the required current.

This schematic should also work for Intel's Programmable Peripheral Interface (PPI) or other output ports. But if you use a port that is always configured as an output, you may want to omit the second inverter by connecting the alarm to pin 2, or use a non-inverting driver like the 7417 or 7407. (A 7406 could also be used instead of the 7416.) The unused gates in the package are available for other uses.

A suggested time for the tone length in the character beep application is approximately 60 milliseconds (ms). The device takes approximately 40 ms to start producing an audible tone, so 60 ms gives start-up time plus a 20 ms tone. For longer applications, the start-up time will probably be negligible. If the alarm vibrates in the IC socket, just put a rubber band around the socket.

This simple project should extend the usefulness and enjoyment of your system. Properly used, it can go a long way towards better human engineering of any system. And now that you have the bell, you'll have to keep your eyes open for a ten dollar whistle project. ■



Parts List

A1 — Dip-Alarm (trademark of Projects Unlimited). Available from KA Electronic Sales, 1220 Majesty Dr., Dallas TX 75247 ($7.95).
IC1 — 7416 TTL integrated circuit — hex inverter buffer/driver.
Misc. — 14-pin DIP socket for 7416 (optional, but recommended).
16-pin DIP socket for alarm (optional, but recommended).

Fig. 1. Schematic and parts list.

Announcing a Computer that thinks in BASIC for only $298.

Ohio Scientific's new Model 500 computer has full mini-computer BASIC in ROM and a minimum of 4K RAM for user programs. BASIC is always there when the computer is turned on.

Make your terminal intelligent!

If your company or school has remote computer terminals, consider the Model 500-1. It fits in the communications line between the terminal and modem or system. When the 500-1 is off, the terminal talks to the modem. When it is on, the terminal talks to the Model 500. Use the Model 500 for short and medium sized programs in BASIC. Use its *immediate* mode as an ultra-powerful scientific calculator!

Computers for Students

Couple a Model 500 to a low cost purchased or leased terminal such as the popular ASR-33 teletype for student instruction. The Model 500's BASIC is a perfect instructional language for students. Since the computer's machine code is also accessible, the Model 500 can challenge the advanced student with other optional software such as our interactive Assembler/Editor.

Personal Computing

The Model 500 has eliminated all of the grief of the first generation personal computers. This computer comes fully assembled, tested, and guaranteed.

Most importantly, it is *easy to use* since the BASIC is always there.

Flexibility and Expandability

The Model 500 uses Ohio Scientific's standard 48 line bus so that it is fully compatible with our Model 400 kits and Challenger™ product lines. Any of thirteen accessory boards including RAM, PROM, parallel, serial, A/D, D/A, cassette, and video graphics can be added as well as peripherals including floppy disks, line printers, color graphics, and more.

The Model 500 is offered here in three forms to meet every application and pocket book.

Specifications

Model 500 Board $298.00

8″ x 10″ fully assembled board including 6502 microprocessor running at 1MHz, 512 bytes of PROM, 8192 bytes of ROM containing 8K BASIC and 4096 bytes of RAM for user programs. The board contains an ACIA based serial interface which is jumper selectable for RS-232C or 20ma loop at 110, 300, 1200, 2400, or 4800 baud.

The 8K BASIC features: full floating point math including transcendental functions, N dimension arrays, multiple letter variables, full string functions, logical operators, PEEK, POKE, USR, and lots more!

The Model 500 board requires +5 volts at 2 amps, −9 volts at 500ma, an external reset switch, and an ASCII serial terminal for operation.

Model 500-1 $429.00

Fully enclosed 500 board with power supply, reset switch, and two 25 pin EIA standard terminal connectors for loop through operation. The 500-1 is 12″ x 15″ x 4″.

Model 500-8 $629.00

This unit is a 500 board in an eight slot Challenger™ case allowing seven slots for expansion. The unit has a heavy-duty UL recognized power supply and is 15″ x 17″ x 10″.

ORDER FORM

Order directly from:
Ohio Scientific; Hiram, Ohio 44234
This introductory offer expires August 31, 1977. Please allow up to 60 days for delivery.

PRINT OR TYPE CLEARLY.

Name ____________

Shipping Address ____________

City ____________ State ______ Zip ______

Payment by: BAC (VISA)____ MC____ Money Order____

Credit Card Account # ____________ Expires ______

Interbank # (Master Charge) ____________

_____ Model 500 Boards @ $298.00 ____________

_____ Model 500-1 Computer @ $429.00 ____________

_____ Model 500-8 Computer @ $629.00 ____________

_____ Additional 4K RAM Memories for 500-8 @ $149.00 ____________

4% Sales Tax (Applies to Ohio Residents only) ____________

TOTAL CHARGED OR ENCLOSED ____________

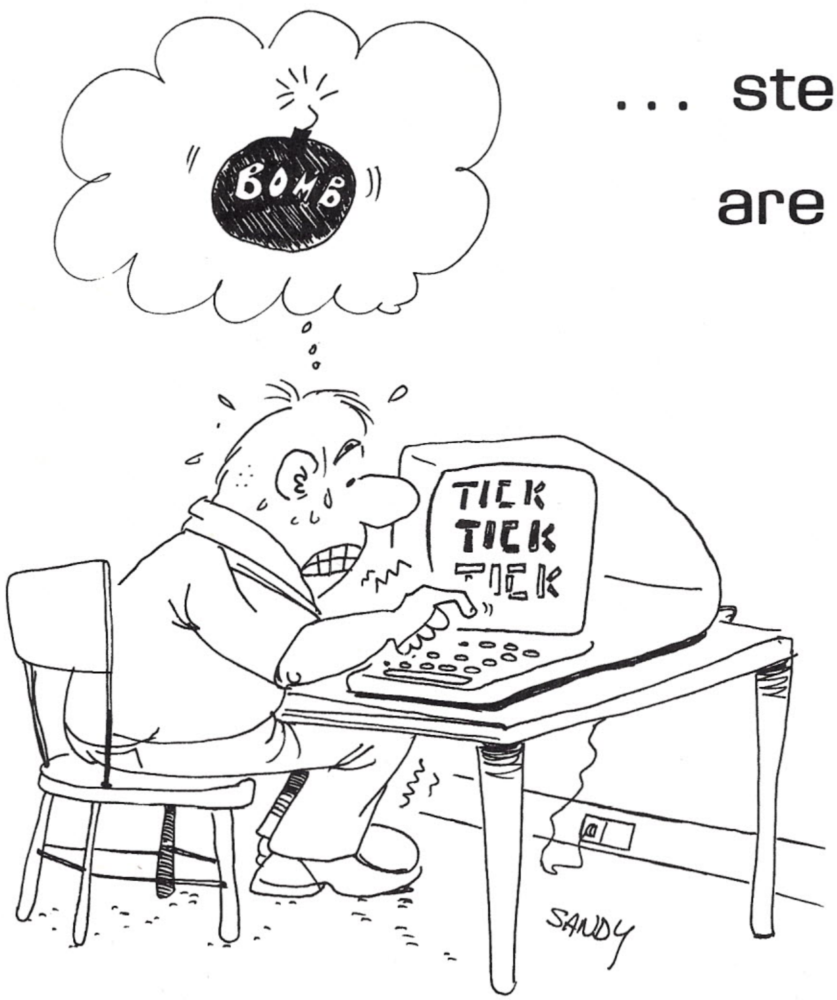
All orders shipped insured UPS unless otherwise requested.

OHIO SCIENTIFIC
Hiram, Ohio 44234

01

Time Bomb Game

... steady nerves are required



Dave Culbertson
238 Exchange St.
Chicopee MA 01013

I wanted a programming technique which would create a more realistic simulation of a real situation. Everyone who has run a Lunar Lander (*101 Basic Games*) simulation finds that they may become so engrossed in their calculations that they forget to depress the return key. If Lunar Lander were written using the technique I describe in this article, you would not have to depress the return key. If you did not key in anything, you would fall due to gravity and crash anyway. This is what would happen in a real situation and so it should happen in your games and simulations.

I am not going to rewrite Lunar Lander but I will introduce a simulation that few of us would care to actually participate in. My game is "Time Bomb." The object is to cut wires in the bomb with the intent to defuse it. If you do not cut any wires, the bomb will either explode or indicate that it is a dud. The results of each of your 20 allowed tries will be displayed on the screen and, at the game's end, comments will show how well you performed.

I have written this game in Mits 8K BASIC, for use with a modified SWTP CT-1024 video terminal, running at 1200 baud, 64 characters per line, with the computer controlled cursor board. I have included, later in this article, suggested program changes for Teletype and scrolling terminal use. There are three special commands used in this program:

OUT 17,19 – Instructs the terminal to "Home Up."

OUT 17,22 – Instructs the terminal to "Erase To End Of Frame."

INP (17) – Fetches the last key enclosure that was input

from Port 17.

After the initial instructions are presented, my terminal shows a display of running numbers in the upper left-hand corner of the screen. This is the visual display of the bomb timer. You must cut the red, green, or blue wires by depressing the R key for red, G key for green, or the B key for blue. The numbers will only run to 100 and then the bomb will detonate or indicate a dud. As you cut the wires, comments will be displayed below the running numbers. One wire will defuse the bomb, one will do nothing, and one will detonate the bomb. If you depress any other key except the space bar, this is a mistake and is taken to mean you wish to cut the purple wire. All purple wires detonate the bomb and the display will show you the ASCII number of the purple wire you cut. The space bar represents removing your hand from the bomb.

The technique I employ to achieve this realistic simulation is to sample the input from Port 17 while the game is running and to act conditionally upon it. You can make all your games much more fun by doing this. You can also get more players involved at several terminals (a time-sharing situation!) and, if they do nothing, you can still act in your part of the game instead waiting for them (a wait statement).

I hope some of you will employ this technique in your games and send them in for publication.

The : symbol permits multiple statements per line. If your BASIC does not permit this, type new lines between the ones listed in the program. If you are using a Teletype, change line #110 to Z=INP(17):PRINTX; and line #500 to NEXT:PRINT:-GOTO 70. If you are using a scrolling terminal, change line #110 to Z=INP(17): (Insert Screen Erase):PRINT X; and change line #500 to NEXT: (Insert Screen Erase):GOTO 70. ■

```
1 REM—WRITTEN BY DAVID C. CULBERTSON
10 PRINT "THE TIME BOMB GAME—YOU HAVE ABOUT 5 SECONDS TO READ THIS."
20 PRINT "THE BOMB WILL TICK TO 100 MAX.—THERE ARE THREE WIRES TO CUT."
30 PRINT "PRESS KEY R FOR RED, G FOR GREEN, OR B FOR BLUE."
40 PRINT "THE SPACE BAR TAKES YOUR HAND OUT OF THE BOMB."
50 PRINT "ALL OTHER KEYS ARE PURPLE WIRES-VERY LETHAL."
60 FOR Q=1 TO 4000:NEXT:OUT 17,19:OUT 17,22
70 Y=INT(100*RND(3))+50:E=INT(3*RND(3))+1
80 F=INT(3*RND(3))+1
90 IF E=F GOTO 80
100 FOR X=O TO 100
110 Z=INP(17):OUT 17,19:PRINT X
120 IF Z≠13 AND Z<32 THEN B1=1:GOTO 340
130 IF Z=82 THEN Z=1:GOTO 180
140 IF Z=71 THEN Z=2:GOTO 230
150 IF Z=66 THEN Z=3:GOTO 280
160 IF Z>32 THEN B1=1:GOTO 340
170 GOTO 340
180 IF Z1=2 GOTO 370
190 Z1=Z:PRINT "SNAP..RED WIRE CUT."
200 IF Z1=E THEN B1=1:GOTO 350
210 IF Z1=F GOTO 330
220 GOTO 340
230 IF Z2=2 GOTO 370
240 Z2=Z:PRINT:PRINT "SNAP,GREEN WIRE CUT."
250 IF Z2=E THEN B1=1:GOTO 350
260 IF Z2=F GOTO 330
270 GOTO 340
280 IF Z3=2 GOTO 370
290 Z3=Z:PRINT:PRINT "SNAP,.BLUE WIRE CUT."
300 IF Z3=E THEN B1=1:GOTO 350
310 IF Z3=F GOTO 330
320 GOTO 340
330 PRINT "######## SUCCESS, THIS BOMB DEFUSED.":PRINT:G1=1:GOTO 390
340 IF B1=1 THEN PRINT "YOU CUT PURPLE WIRE # ";Z
350 IF B1=1 THEN PRINT:PRINT "******** BANG.":GOTO 390
360 IF X=Y THEN B1=1:GOTO 350
370 NEXT X
380 PRINT:PRINT "THIS BOMB WAS A DUD-----WHEW!! LUCKY!":D1=1
390 D=D+D1:G=G+G1:T=T+1:B=B+B1
400 Z1=0:Z2=0:Z3=0:G1=0:D1=0:B1=0
410 PRINT "ATTEMPTS","EXPLOSIONS","DEFUSIONS","DUDS"
420 PRINT T,B,G,D
430 PRINT:IF T=20 GOTO 510
440 PRINT "YOU HAVE 5 SECONDS. I HEAR A BOMB TICKING."
450 PRINT:PRINT "%%%%%%%% PRESS THE SPACE BAR OR I'LL CUT A WIRE%%%%%%%%.":PRINT
460 FOR F1=1 TO 350
470 IF F1=100 THEN PRINT "TICK",
480 IF F1=200 THEN PRINT "TICK",
490 IF F1=300 THEN PRINT "TICK"
500 NEXT:OUT 17,19:OUT 17,22:GOTO 70
510 IF B=0 AND G>15 THEN PRINT "A FANTASTIC PERFORMANCE"
520 IF B<5 AND G>10 THEN PRINT "A GOOD TRY, BUT NOT GREAT."
530 PRINT "YOU WERE BOMBED" B;"TIMES,YOU DEFUSED" G,"BOMBS"
540 IF D>1 THEN PRINT "BUT YOU DID GAMBLE WITH" D;"DUDS"
550 PRINT:PRINT "WHY DON'T YOU TRY AGAIN SOMETIME!"
560 END
```

Program Listing.

Randy Miller
1010 East Lemon #5
Tempe AZ 85281

Try a Do-All Program!

... it will even balance your checkbook

The next time a friend comes over and you want to show off the ol' computer, why not show how useful the computer can be for simple everyday chores around the house? Wouldn't it be nice to have the computer keep track of names, addresses and phone numbers? And how about a list of the books in your library in order of title and in order by author? How about a list of your record collection grouped by subject matter, type of music, performers, or whatever? And how about an index of all the articles from

all seventeen of the magazines you subscribe to, grouped both by subject and in alphabetical order?

This little program described here is what you call your old-fashioned all-American super-duper all-around top-notch all-incorporated expandable extra-special does-all program. It will keep all of the lists mentioned above plus your checking account, accounts receivable, accounts payable, inventory, and anything else you might come up with.

What is It?

First of all, there's very few of us rich enough to own a floppy disk drive. So this program is designed for lists of limited length and no floppy is needed.

The program simply uses four lists of data. Two lists are numerical and two are lists of string data. The two numerical lists are contained in matrix N, and the list of strings comprise matrix A$. What we call an *entry* in the list is a collection of the four data items. Not all four items need be used. For instance, a list of phone numbers might use one string for the number and one string for the name. The two numerical items would just remain zero.

The list of data items may be sorted at any time and put in order. The sort can be done using any of the four data items as sorting criteria.

Operation

After loading BASIC and putting in the program, set the variable MAX in line 1100 to the maximum number of entries that the system will be limited to. It's now set to 100 entries.

The subscripts can be adjusted in line 1080 to accommodate a different capacity of names. Just change the 100 to any other number. That number, however, must be greater than or equal to the value assigned to MAX in line 1100. The only limit to the number of entries allowed is the amount of

```
COMMAND? S

 1 AMOUNT
 2 INVOICE #
 3 NAME
 4 DATE

TYPE NUMBER CORRESPONDING TO DATA TO BE SORTED BY? 3

COMMAND? P
COMPLETE PRINT OR PARTIAL PRINT (C OR P)? C

---------------------------------------------------------------

AMOUNT       INVOICE #         NAME                   DATE
 120.43      88703             BANK LOAN #271         3/02/77
 1054.32     3422              JOE'S JUNKYARD         2/01/77
 654         4022              JOE'S JUNKYARD         2/07/77
 20.76       0                 MR. CARSON             2/25/77
 234.98      12003             SMITH ELECTRONICS      1/03/77
 109.32      12190             SMITH ELECTRONICS      1/23/77
 3.09        13155             SMITH ELECTRONICS      3/10/77

---------------------------------------------------------------

COMMAND? S

 1 AMOUNT
 2 INVOICE #
 3 NAME
 4 DATE

TYPE NUMBER CORRESPONDING TO DATA TO BE SORTED BY? 1

COMMAND? P
COMPLETE PRINT OR PARTIAL PRINT (C OR P)? P

 1 AMOUNT
 2 INVOICE #
 3 NAME
 4 DATE

TYPE IN THE NUMBER CORRESPONDING TO THE DATA COLUMN
FOR WHICH TO SPECIFY LIMITS? 1
ENTER MIN,MAX? 200,5000

---------------------------------------------------------------

AMOUNT       INVOICE #         NAME                   DATE
 234.98      12003             SMITH ELECTRONICS      1/03/77
 654         4022              JOE'S JUNKYARD         2/07/77
 1054.32     3422              JOE'S JUNKYARD         2/01/77

---------------------------------------------------------------

COMMAND? S

 1 AMOUNT
 2 INVOICE #
 3 NAME
 4 DATE

TYPE NUMBER CORRESPONDING TO DATA TO BE SORTED BY? 4

COMMAND? P
COMPLETE PRINT OR PARTIAL PRINT (C OR P)? P

 1 AMOUNT
 2 INVOICE #
 3 NAME
 4 DATE

TYPE IN THE NUMBER CORRESPONDING TO THE DATA COLUMN
FOR WHICH TO SPECIFY LIMITS? 4
INPUT STARTING STRING, ENDING STRING? 1/01/77,2/15/77

---------------------------------------------------------------

AMOUNT       INVOICE #         NAME                   DATE
 234.98      12003             SMITH ELECTRONICS      1/03/77
 109.32      12190             SMITH ELECTRONICS      1/23/77
 1054.32     3422              JOE'S JUNKYARD         2/01/77
 654         4022              JOE'S JUNKYARD         2/07/77

---------------------------------------------------------------

COMMAND?
```

Example 1.

```
COMMAND? S

 1 PRICE
 2 PART NO.
 3 ITEM
 4 —

TYPE NUMBER CORRESPONDING TO DATA TO BE SORTED BY? 3

COMMAND? P
COMPLETE PRINT OR PARTIAL PRINT (C OR P)? C

-----------------------------------------------------------------

PRICE           PART NO.             ITEM                         —

 1.1            231199          GADGET — LITTLE BIGGER      —

 .98            231198          GADGET — RATHER SMALL       —

 3.55           33823           THING 12 X 2                —

 4.98           33881           THING 12 X 2.5              —

 3.55           22651           WHATSIT — MED               —

 4125.35        22652           WHATSIT — PRETTY BIG        —

 2.98           22649           WHATSIT — SM                —

-----------------------------------------------------------------

COMMAND? S

 1 PRICE
 2 PART NO.
 3 ITEM
 4 —

TYPE NUMBER CORRESPONDING TO DATA TO BE SORTED BY? 2

COMMAND? P
COMPLETE PRINT OR PARTIAL PRINT (C OR P)? C

-----------------------------------------------------------------

PRICE           PART NO.             ITEM                         —

 2.98           22649           WHATSIT — SM                —

 3.55           22651           WHATSIT — MED               —

 4125.35        22652           WHATSIT — PRETTY BIG        —

 3.55           33823           THING 12 X 2                —

 4.98           33881           THING 12 X 2.5              —

 .98            231198          GADGET — RATHER SMALL       —

 1.1            231199          GADGET — LITTLE BIGGER      —

-----------------------------------------------------------------

COMMAND?
```

Example 2.

Program listing (continued on following pages).

```
LIST

1000 REM          ***** FILE MANAGEMENT *****
1010 REM              LOOK MOM, NO FLOPPIES!
1020 REM
1030 REM                BY RANDY MILLER
1040 REM                1010 E. LEMON #5
1050 REM             TEMPE, ARIZONA 85281
1060 REM
1070 CLEAR 1000
1080 DIM N(2,100),A$(2,100)
1090 CT$="LDSPARX"
1100 MAX=100
1110 P=0
1120 PRINT "DATA MANAGEMENT MINUS FLOPPIES"
1130 PRINT
1140 GOSUB 9000
1150 INPUT "COMMAND";C$
1160 FOR J=1 TO LEN(CT$)
1170 IF C$=MID$(CT$,J,1) THEN 1210
1180 NEXT J
1190 PRINT "WHAT?"
```

memory you have.

When creating a list of entries for the very first time, use the Load (L) command (see below). After the list is once entered it may be sorted in order by any of the four data items by using the Sort (S) command. Printouts are obtained by using the Print (P) command. Entries can be added or removed from the list by the Add (A) or Remove (R) command. Before turning off the computer use the Dump (D) command to save the list on tape. The next time you use the program with that list you'll need only to run the tape through using the Load (L) command.

Here is a closer look at the commands.

Commands

(L) Load. One use of this command is for entering a list for the very first time. The titles for each of the four data items will be set at this time. Simply type in each data item; the two numerical items come first, then the two strings items. Use a separate line for each data item. If an entry was typed in wrong, the only way to correct it is to use the Remove command after the list is completely entered and use the Add command to put the correct entry in.

After the very last entry is made you need to make one more entry consisting of two zeros for the numerical data and a $ for each of the string items. This is the information that tells the computer the list is done. After that, use the Add and Remove commands to make any adjustments to the list.

Another use of the Load command is to input an entire list of entries from tape that was previously made using the Dump command.

(D) Dump. This command can be used if the terminal being used has some method of producing a machine-readable copy of what's printed, such as the paper-tape punch on an ASR Tele-

type. The program will stop at this point. This is to allow you to set the nulls in the BASIC to at least six. Then type in CONT, turn on the tape punch and hit return. A leader will be punched followed by the whole list. Then use the Load command at any time to read the list back in.

(A) Add. Simply type in the four data items that comprise the entry in the proper order. The entry will be placed at the end of the existing list but using the Sort command will place it in the proper sequence with the other entries.

(R) Remove. Just like the Add command except that an existing entry is removed from the list.

(S) Sort. The entire list of entries can be sorted in order. Simply specify which column of data is to be used as criteria for sorting and allow it a few seconds to do the sorting.

(P) Print. The entire list or just a portion of it can be printed. If a partial listing is specified, you must indicate which column is to be used to determine the limits of the printing. But remember – the list must be sorted by that column of data items before a partial print is specified. For instance, let's say you have a list of accounts receivable. One numerical data items is the amount the customer owes; the other is the length of time the amount has been owed. One string data item is the account's name and the other string is not used. There are several ways that a partial print can be used in this case. Perhaps you want a list of all accounts that owe more than $150. First you must sort the list using that column of data items, then specify a partial print using limits on that column starting at 150 and however high you want. Or you might sort the list by the second numerical data (number of weeks old). Then you can get a printout of all accounts more than 10 weeks old by specifying limits on

```
1200 GOTO 1150
1210 ON J GOTO 2000,3000,4000,5000,6000,7000,8000
1220 GOTO 1190
2000 REM - - 'L' (LOAD) COMMAND - -
2010 INPUT "TITLE FOR 1ST NUMERICAL DATA";T$(1)
2020 INPUT "TITLE FOR 2ND NUMERICAL DATA";T$(2)
2030 INPUT "TITLE FOR 1ST STRING DATA";T$(3)
2040 INPUT "TITLE FOR 2ND STRING DATA";T$(4)
2050 P=1
2060 PRINT
2070 INPUT N(1,P),N(2,P)
2080 INPUT A$(1,P)
2090 INPUT A$(2,P)
2100 IF N(1,P) <> 0 OR N(2,P) <> 0 THEN 2120
2110 IF A$(1,P) = "$" AND A$(2,P) = "$" THEN 1140
2120 P=P+1
2130 IF P <= MAX THEN 2060
2140 GOSUB 9050
2150 GOTO 1140
3000 REM - - 'D' (DUMP) COMMAND - -
3010 PRINT "TYPE 'NULL 6' THEN TYPE 'CONT' "
3020 STOP
3030 GOSUB 9520
3040 FOR J=1 TO P-1
3050 PRINT N(1,J);",";N(2,J)
3060 PRINT A$(1,J)
3070 PRINT A$(2,J)
3080 NEXT J
3090 REM - - PRINT END OF LIST DATA - -
3100 PRINT "0,0"
3110 PRINT "$"
3120 PRINT "$"
3130 GOSUB 9520
3140 PRINT "YOU MAY SET NULLS TO ZERO, THEN TYPE 'CONT' "
3150 STOP
3160 GOTO 1140
4000 REM - - 'S' (SORT) COMMAND - -
4010 GOSUB 9080
4020 INPUT "TYPE NUMBER CORRESPONDING TO DATA TO BE SORTED BY";T
4030 IF T > 2 THEN 4130
4040 REM - - BUBBLE SORT FOR NUMERICAL DATA - -
4050 K=P-2
4060 FOR J=1 TO K
4070 IF N(T,J) <= N(T,J+1) THEN 4090
4080 GOSUB 9210
4090 NEXT J
4100 K=K-1
4110 IF K >= 1 THEN 4060
4120 GOTO 1140
4130 REM - - BUBBLE SORT FOR STRING DATA - -
4140 T=T-2
4150 K=P-2
4160 FOR J=1 TO K
4170 IF A$(T,J) <= A$(T,J+1) THEN 4190
4180 GOSUB 9210
4190 NEXT J
4200 K=K-1
4210 IF K >= 1 THEN 4160
4220 GOTO 1140
5000 REM - - 'P' (PRINT) COMMAND - -
5010 INPUT "COMPLETE PRINT OR PARTIAL PRINT (C OR P)";C$
5020 IF C$="P" THEN 5100
5030 GOSUB 9000
5040 GOSUB 9350
5050 FOR J=1 TO P-1
5060 GOSUB 9410
5070 NEXT J
5080 GOSUB 9460
5090 GOTO 1140
5100 REM - - PARTIAL PRINTOUT - -
5110 GOSUB 9080
5120 PRINT "TYPE IN THE NUMBER CORRESPONDING TO THE DATA COLUMN"
5130 INPUT "FOR WHICH TO SPECIFY LIMITS";T
5140 IF T > 2 THEN 5230
5150 INPUT "ENTER MIN,MAX";L,H
5160 GOSUB 9000
5170 GOSUB 9350
5180 FOR J=1 TO P-1
5190 IF N(T,J) >= L AND N(T,J) <= H THEN GOSUB 9410
5200 NEXT J
5210 GOSUB 9460
5220 GOTO 1140
5230 T=T-2
5240 INPUT "INPUT STARTING STRING, ENDING STRING";B1$,C$
5250 IF C$ < B1$ THEN 5240
5260 GOSUB 9000
5270 GOSUB 9350
5280 FOR J=1 TO P-1
5290 IF A$(T,J) >= B1$ AND A$(T,J) <= C$ THEN GOSUB 9410
5300 NEXT J
5310 GOSUB 9460
```

that column of data.

(X) Custom. The program is constructed to allow you to set up your own routine for the command called X. Just insert the routine for the command at line 8010 and have it GOTO 1150 when finished. The only restriction in programming is to keep the variable P equal to *one more* than the number of entries in the list.

Notes on Sorting

Let's say you need to sort the list by the first string data, but when there are many entries with the same value for that string, you want them placed in order by the second numerical data item. And perhaps when many of the entries have those two items the same, you might want those entries sorted in order of the other (second) string data item. To do this, sort the list more than once, starting with the column of information that's least important, ending with a sort on the data items of primary consideration.

Examples of Operation

In these examples, we'll assume that the entries have already been loaded by the Load command. In Example 1, we have a list of accounts payable. The first string item is the name of the creditor; the second string is the date of purchase. The first numerical item is the amount of the purchase and the second item is a reference number (e.g., an invoice or statement number). We want to see the whole list in order of creditor, then see only those bills for more than $200. Finally, we want to see only those bills that are older than 2/15/77. You can see how the various commands are used to sort and print the list.

In Example 2, we have a list of the inventory for a small store. We want first a list of every item in order of the name of the item. Next we want a list in order of part number. ■

```
5320 GOTO 1140
6000 REM -- 'A' (ADD) COMMAND --
6010 IF P<MAX THEN 6040
6020 GOSUB 9050
6030 GOTO 1150
6040 PRINT "ENTER THE FOLLOWING DATA:"
6050 GOSUB 9150
6060 P=P+1
6070 IF LEN(A$(1,P))<25 AND LEN(A$(2,P))<25 THEN 1140
6080 PRINT "STRING TOO LONG - WARNING ONLY."
6090 GOTO 1140
7000 REM -- 'R' (REMOVE) COMMAND --
7010 PRINT "ENTER THE FOLLOWING DATA:"
7020 GOSUB 9150
7030 REM -- FIND ENTRY TO BE DELETED --
7040 FOR J=1 TO P-1
7050 IF N(1,J) <> N(1,P) OR N(2,J) <> N(2,P) THEN 7160
7060 IF A$(1,J) <> A$(1,P) OR A$(2,J) <> A$(2,P) THEN 7160
7070 REM -- FOUND IT, NOW ADJUST REST OF LIST --
7080 FOR K=J TO P-2
7090 FOR T=1 TO 2
7100 A$(T,K)=A$(T,K+1)
7110 N(T,K)=N(T,K+1)
7120 NEXT T
7130 NEXT K
7140 P=P-1
7150 GOTO 1150
7160 NEXT J
7170 GOTO 1150
8000 REM -- 'X' (CUSTOM) COMMAND --
8010 REM -- INSERT ROUTINE HERE --
8020 GOTO 1150
9000 REM -- PRODUCES THREE LINEFEEDS --
9010 FOR J=1 TO 3
9020 PRINT
9030 NEXT J
9040 RETURN
9050 REM -- PRINTS ERROR MESSAGE --
9060 PRINT "ATTEMPT TO EXCEED MAXIMUM NUMBER OF ENTRIES ALLOWED."
9070 RETURN
9080 REM -- PRINTS TITLES OF DATA --
9090 PRINT
9100 FOR J=1 TO 4
9110 PRINT J;T$(J)
9120 NEXT J
9130 PRINT
9140 RETURN
9150 REM -- PRINTS TITLES OF DATA AND ALLOWS INPUT --
9160 FOR J=1 TO 4
9170 PRINT T$(J)
9180 NEXT J
9190 INPUT N(1,P),N(2,P),A$(1,P),A$(2,P)
9200 RETURN
9210 REM -- BUBBLE SORT SWAP --
9220 X1=N(1,J+1)
9230 X2=N(2,J+1)
9240 B1$=A$(1,J+1)
9250 B2$=A$(2,J+1)
9260 FOR X=1 TO 2
9270 N(X,J+1)=N(X,J)
9280 A$(X,J+1)=A$(X,J)
9290 NEXT X
9300 N(1,J)=X1
9310 N(2,J)=X2
9320 A$(1,J)=B1$
9330 A$(2,J)=B2$
9340 RETURN
9350 REM -- PRINTS TITLES --
9360 GOSUB 9460
9370 PRINT T$(1);TAB(13);T$(2);
9380 PRINT TAB(28);T$(3);TAB(52);T$(4)
9390 PRINT
9400 RETURN
9410 REM -- PRINTS ONE ENTRY SPECIFIED BY J --
9420 PRINT N(1,J);TAB(13);N(2,J);
9430 PRINT TAB(25);A$(1,J);TAB(49);A$(2,J)
9440 PRINT
9450 RETURN
9460 REM -- PRINTS A LINE OF -'S --
9470 FOR J=1 TO 70
9480 PRINT "-";
9490 NEXT J
9500 PRINT:PRINT
9510 RETURN
9520 REM -- PUTS LEADER ON TAPE --
9530 FOR J=1 TO 50
9540 PRINT CHR$(0);
9550 NEXT J
9560 RETURN
9999 END
OK
```

Sooo, You Want to be an Author

George Young
Sierra High School
Tollhouse CA 93667

John Craig and Wayne Green have been asking you in every subtle (and some not so subtle) way that they can think of to write articles for OUR magazines, *73* and *Kilobaud*. I say our magazines not because I own stock in either, but because they make me feel that I am a member of the *73/Kilobaud* family. I managed to have my very first article published in *Kilobaud* #5 and I'll have a little article called a filler coming up shortly in *73*.

Let Me Count the Reasons

So let me tell you why you should write for these magazines and then I will try to pass on what I have learned so far about the business of writing for a national magazine. Perhaps I can convince you that us ordinary people can become "famous authors".

Money! (#1). The first and perhaps most important reason that you should write for *73* and/or *Kilobaud* is that they will pay you for your article. You won't get rich, but the magazines pay, and they pay well. And most important, they pay promptly for your material. With the money that you make from your article, you can add to your station equipment or computer system. In fact, you might be clever enough to get *73* or *Kilobaud* to finance your whole set-up. I'll clue you in on this move a little later.

Fame (#2). Second, you should write for the fame that getting published brings. It really is just a little frightening at first. Getting published in a national magazine puts your name where a lot of people see it. I found myself being introduced in March at the Central Valley Computer Club meeting as "that famous Kilobaud author". I had not even been in print yet. It was embarrassing.

Each month I grew more apprehensive when I saw the high calibre of the articles appearing in *Kilobaud* and I did not see how my puny efforts would ever measure up to the standards being set by the other authors. Heck, just because you get something published in a magazine doesn't make you an expert. Maybe some of the authors you see in the magazines are experts, but I suspect that there are more ordinary people writing for *73/Kilobaud* than there are experts.

Large readership (#3). You are going to be read. After all, there isn't much point in writing anything if no one is going to read it. As long as you are going to go to the trouble of writing, it might just as well be for the magazines that have the greatest circulation.

The learning process (#4). You will learn something new. I think that it is human nature to be insatiably curious. When we stop learning, we most likely should be checked to see if we also stopped breathing. I enjoy learning new things, new processes, new procedures. I suspect that if you haven't stopped breathing, you do also.

How to win friends and influence people (#5). You will make a lot of new friends. When I first got my amateur radio license, one of the things that most impressed me was the vast number of new friends that I made in a very short time. This same thing is happening all over again in the field of microcomputers.

The creative urge (#6). All of us have the urge to create. If we are of the appropriate age, we create replicas of ourselves. If we are a few years past this stage, we build things. We create furniture in the home wood shop. We paint pictures. We take photographs. We build amateur radio equipment. We build computers and write software. This creative urge is built into us at conception and we never lose it. Writing is one way to satisfy this drive and if someone will pay you for it, so much the better.

Now For the Excuses . . .

What keeps you from writing? Here are some sample replies: "Everything has already been done. All I know has already been written and published by someone else. There is nothing left for me to write about. I'm not good enough (smart enough, clever enough, etc.) to write anything: I can't even write a decent letter to a friend."

I'm sure that each one of the above sounds familiar to you. You have probably used one or more of these excuses. Or the one we all use — I don't have the time to write an article; I'm too busy. Bunk! Let's take a look at some of those excuses and see just how much of each one is true.

"Everything has already been done". If it has, then mankind will make no further progress throughout the remainder of time. If you believe that everything has already been done, then go make the down-payment on that gravesite that you've been contemplating.

"All that I know has already been written and published by someone else." This is entirely possible. But it shouldn't stop you for much more than 15 minutes. Look at my articles in *Kilobaud*. Every circuit has already been published. Every idea belongs to another author. All I'm doing is

presenting the same old material in a different light. And look what yours truly is doing right now! I'm simply repeating the material that John and Wayne have already written. The only thing they didn't give you was reason #6 which I mentioned earlier.

Look at the articles on how to make PC boards, on power supplies, on electronic keyers. All have been "peated" and repeated. Yet they still publish more on these subjects. Why? All it takes to get an old idea back in print is a new or different slant. So even if everything has already been written, published, and forgotten, you can't use that excuse any more.

"I'm not good enough, smart enough, etc. . . " Don't sell yourself short. Who told you so? What expert did you consult? Write an article, send it in, and see if you are as rotten at writing as you think you are.

The First Attempt

The first manuscript I submitted was written in pencil on graph paper. I printed it using all capital letters. I left no margins. I left no space between lines. In spite of this, the article went from Peterborough to John Craig. I imagine that John shuddered when he saw that first manuscript. He thought enough of it to write me a very nice letter, and talk me into typing the manuscript on plain white paper, using both upper and lower case letters (instead of all capitals), double spacing and leaving a wide margin. He still doesn't know that he accomplished a minor miracle since I do not type, and when I do it could only be described as terrible. He got the typewritten manuscript, made some changes and suggestions, and sent it back. After several trips the manuscript didn't come back; an acceptance letter came back instead. John put an awful lot of work into that manuscript, and into almost every one of the subsequent manuscripts as well. You have him to thank for the smoothness of the Kilobaud Klassroom series.

Suppose you get rejected. If John Craig has to write a letter of rejection, I'll bet that he will let you down so easy that you will turn right around and write something else for him. You can write for a lot of editors before you find another with John's skill.

What to Write About

And what should you write about? Anything. It should be something that you have done or a problem you have solved. For *73* or *Kilobaud*, it should be something practical. The magazines are aimed at practical solutions to real problems by ordinary people. Whatever you've done, whatever problems you have solved, whatever bugs you've ironed out of equipment . . . that is what you should write about.

How To Do It

Double Spaced. Ordinary plain paper. Wide margins. Typewritten. Submit as many photographs as you can. You get paid for them also. If you have to hire a pro to take the pictures, do it. If you have a friend who is a ham or a computer phreak, get his help. Polaroids are a definite no-no. All photos should be sharp, crisp, glossy 8 x 10's (or 4 x 5's). You could, however, take preliminary Polaroid shots and let John choose which ones he wants. He will get back to you with his choices, then you can get the glossies made of those photos and send them in.

Diagrams and Drawings

You probably aren't the world's best artist, but anyone can submit a neat drawing. All it takes is a little TLC (Tender Loving Care) and a *template*. Each drawing goes on a separate sheet of paper, not into the text. The reason for this is that the drawings are sent out to be reproduced by professional draftsmen, while the text stays in the office for editing, typesetting and layout. A separate sheet of paper should be included that lists the captions you want on each drawing. I have been putting captions on the drawings as well as providing the caption sheet. A caption sheet is needed so the type can be set even though the figures have been sent out for drafting.

People often quote the old saying "a picture is worth a thousand words". I don't know if a picture will save you a thousand words of typing, but I do know that without pictures or diagrams, your article will have to be superb to get published. Both pictures and diagrams will help immeasurably.

An Example

Writing an article changes the way you do things a little. Let's suppose that you just bought a computer in kit form. You think, after reading all this that maybe you can get *73/Kilobaud* to pay for your kit. (I told you earlier that I have a plan for you to get someone to finance your computer.) To manage this, you are going to have to write up the construction procedure and get it published. So instead of unpacking the kit, glancing through the instructions, and heating up the soldering iron, a bit of pre-planning is needed.

First, take the camera and get some shots of the raw kit as it arrives. Readers will want to know if the parts were well packed, and if the *show* parts such as the front panel were well protected. How well a manufacturer protects his equipment during shipping is a direct reflection of how much he thinks of it. So take lots of pictures as you unpack, and hope that the editor will buy at least one. Watch the focus; a blurry picture is worthless. And keep in mind (if you're not a photographer) that there are many amateur photographers out there who will gladly take your photos in exchange for a credit line in the magazine.

Keep careful notes as you proceed. Look at the quality of the components, the circuit board, the paint job. Jot down your impressions.

Note any problems. Readers will be particularly interested here. If you had a really rough time with this particular kit, then tell it like it is. That's what *73* and *Kilobaud* are all about. If you got stung, then let it all hang out. If it is a serious problem, John will send a copy of the manuscript to the kit manufacturer for a response (which will also be published). You want results . . . this will get results that a thousand letters wouldn't produce.

Get pictures of the completed kit. It should look as good as those published in the manufacturer's ads. If it doesn't, then you definitely have to have that picture for your article. But be fair . . . if your completed project looks poor because you dropped it on the floor during construction, you can't lay that on the kit manufacturer. If something like this happened, write the kit manufacturer for a photo of one that didn't get dropped so you can give the right impression in your article.

Summary

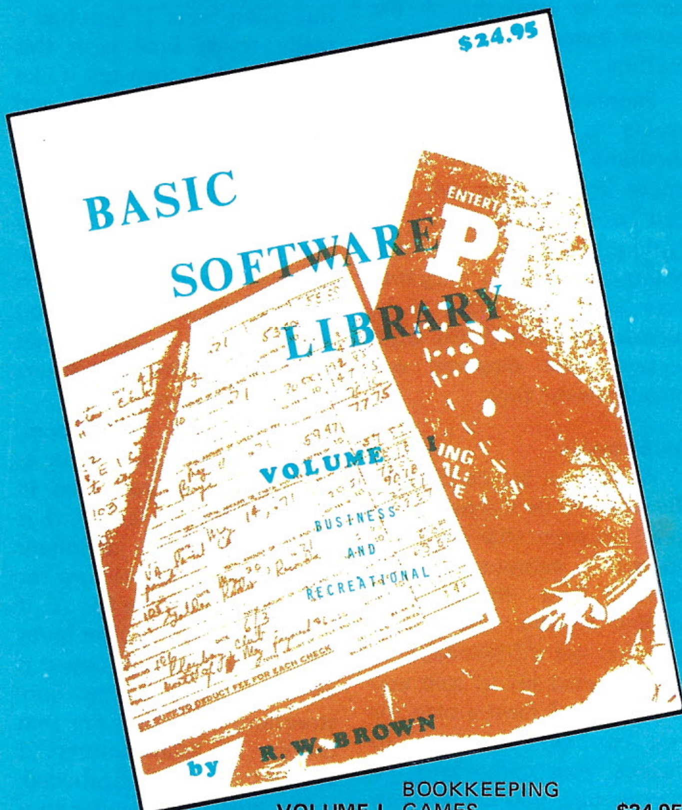
Write. You don't have time? Knock off watching TV for two hours each evening for a week. Put the priorities where they belong. You can't type? Join the group; neither can I. I write because I enjoy it. I am having more fun writing than anything I have done in a long time.

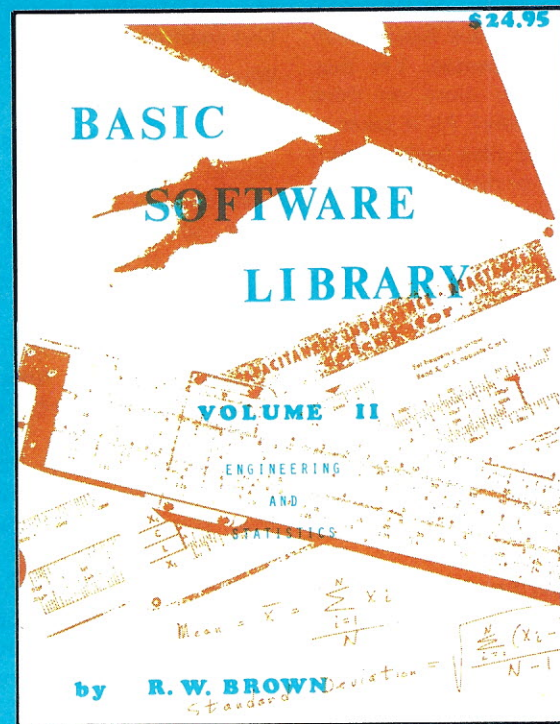
I am satisfying a creative urge and darned if somebody isn't paying me good money to do it. Got a little idea? Write up a filler. Write one typewritten page manuscript and send it to *73* or *Kilobaud*, Peterborough, New Hampshire 03458. Get your feet wet. Make 'ol John Craig earn his keep. Try it, you'll like it. ■

The BASIC SOFT

The "HOW-TO-DO IT" Books for the "DO-IT" Person

Written in compatible BASIC immediately executable in ANY computer with at least 4K, NO other peripherals needed.





VOLUME I BOOKKEEPING / GAMES / PICTURES . . $24.95

VOLUME II MATH & ENGINEERING / PLOTTING & STAT / BASIC STATEMENT DEF . . $24.95

ARIZONA

Byte Shop of Phoenix
Tempe, AZ 85281
(602) 894-1129

Byte Shop of Phoenix-West
Phoenix, AZ 80529
/602) 942-7300

Desert Data Computer Store
Tucson, AZ 85702

CALIFORNIA

A-VID Electronics
Long Beach, CA 90806
(213) 426-5526

Byte Shop of Berkeley
Berkeley, CA 94703
(415) 845-6366

Byte Shop of Campbell
San Jose, CA 95124
(408) 377-4685

Byte Shop of Diablo Valley
Walnut Creek, CA 94596
(415) 933-6252

Byte Shop of Fresno
Fresno, CA 93703

Byte Shop of Hayward
Hayward, CA 94541
(415) 537-BYTE

Byte Shop of Lawndale
Lawndale, CA 90260
(213) 371-2421

Byte Shop of Mt. View
Mt. View, CA 94040
(415) 969-5464

Byte Shop of Palo Alto
Palo Alto, CA 94306
(415) 327-8080

Byte Shop of Pasadena
Pasadena, CA 91101
(213) 684-3311

Byte Shop of Sacramento
Citrus Heights, CA 95610
(916) 726-2557

Byte Shop of San Diego
San Diego, CA 92111
(714) 565-8008

Byte Shop of the San Fernando Valley
Tarzana, CA 93156
(213) 343-3919

Byte Shop of San Jose
San Jose, CA 95123
(408) 226 8383

Byte Shop of San Mateo
San Mateo, CA 94403
(415) 341-4200

Byte Shop of San Rafael
San Rafael, CA 94901
(415) 457-9311

Byte Shop of Santa Barbara
Santa Barbara, CA 93101
(805) 966-2557

Byte Shop of Santa Clara
Santa Clara, CA 95051
(408) 249-4221

Byte Shop/Thousand Oaks
Thousand Oaks, CA 91360
(805) 497 9595

Byte Shop of Westminster
Westminster, CA 92683
(714) 894-9131

Byte Shops, Inc.
Sunnyvale, CA 94086
(408) 734-9000

Byte Shop of Tarzana
Tarzana, CA 91356
(213) 343-3919

The Computer Mart
Orange, CA 92667
(714) 633-1222

Computer Store of San Francisco
San Francisco, CA 94103
(415) 431-0640

The Computer Store
Santa Monica, CA 90401
(213) 451-0713

People's Computer Shop
Sherman Oaks, CA 91423
(213) 789-7514

Computer Components
Van Nuys CA 91411
(213) 786-7411

The Computer Shack
San Leandro, CA 94577
(415) 895-9363

Computerworld Stores
Redding, CA 96001

Upland Computer Labs
Upland, CA 91786
(714) 981-1503

CANADA

Byte Shop of Vancouver
Vancouver 9, B.C.
(604) 736 7221

The Pacific Computer Store
Vancouver, B.C. V5R 2J4
(604) 438-DATA

Trintronics
Toronto, Ontario
(416) 598-0262

COLORADO

Byte Shop/Arapahoe Co.
Englewood, CO 80110
(303) 761-6232

Byte Shop of Boulder
Boulder, CO 80301
(303) 449-6233

The Computer Hut
Denver CO, 80202
(303) 422-7040

CONNECTICUT

The Computer Store
Windsor Locks, CT 06096

FLORIDA

Byte Shop of Cocoa Bch.
Cocoa Beach, FL 32931
(305) 784-1881

Byte Shop of Miami
Miami, FL 33155
(305) 264 BYTE

Computer Hut
Miami Lakes, FL 33014
(305) 821-2667

MicroComputer Systems Inc.
Tampa, FL 33609
(813) 879 4301

Micro Computer Systems & Sales
Pompano Beach, FL 33068
(305) 972-6093

Williams Radio & TV, Inc.
Jacksonville, FL 32206
(904) 354-5460

ILLINOIS

American Microprocessors Equipment & Supply Corp.
Prairie View, IL 60069
(312) 634-0076

INDIANA

Computer Specialists
W. Lafayette, IN 47906
(317) 743-1711

The Data Domain
Bloomington, IN 47401
(812) 334-3607

Graham Electronics
Indianapolis, IN 46204
(317) 634-8202

JAPAN

Byte Shop of Tokyo
2-9-9 Sotokanda
Chiyodaku, Tokyo
Kiyotake Ikeda

The Home Computer Store
Indianapolis, IN 46229
(317) 894-3319

KENTUCKY

The Data Domain
Lexington, KY 40502
(606) 233-3346

MARYLAND

Computer Workshop
Rockville, MD 20852
(301) 468-0455

Science Education Ext. Corp.
Potomac, MD 20854
(301) 299-9506

MASSACHUSETTS

Computer Mart, Inc.
Waltham, MA 02154
(617) 899-4540

MINNESOTA

Byte Shop of Eagan
Eagan, MN 55121
(612) 452-1841

MISSOURI

Computer Systems Center of St. Louis, Inc.
Chesterfield, MO 63017
(314) 576-5020

Computer Workshop
Kansas City, MO 64152
(816) 741 5055

Computer Workshop
Kansas City, MO 64152
(816) 741-5055

NEW HAMPSHIRE

Computer Mart of NH
Nashua, NH 03060
(603) 883-2386

Microcomputers, Inc.
Nashua, NH 03060
(603) 889-1646

NEW JERSEY

The Computer Mart of New Jersey
Iselin, NJ 08830

Hoboken Computer Works
Hoboken, NJ 07030
(201) 420 1644

NEW YORK

Byte Shop of Levittown
Levittown, NY 11756
(516) 731-8116

Computer Mart of Long I
East Meadow, NY 11554
(516) 794-0510

The Computer Mart of NY
New York, NY 10001
(212) 279-1048

Mini Micro Mart
Syracuse, NY 13203
(315) 422-4467

Synchro-Sound Enterprises
Hollis, NY 11423
(212) 468-7067

OHIO

Digital Design
Cincinnati, OH 45243
(513) 561-6733

OKLAHOMA

High Technology
Oklahoma City, OK 73116
(405) 842-2021

OREGON

Byte Shop of Beaverton
Beaverton, OR 97005
(503) 644-2686

Byte Shop of Portland
Portland, OR 97201
(503) 223-3496

PENNSYLVANIA

Byte Shop of Bryn Mawr
Bryn Mawr, PA 19010
(215) 525-7712

Personal Computer Corp.
Frazer, PA 19355
(215) 647-8460

RHODE ISLAND

Computer Power Inc.
Warwick, RI 02886
(401) 738-4477

SOUTH CAROLINA

Byte Shop of Columbia
Columbia, SC 29205
(803) 771-7824

TENNESSEE

Byte'Tronics
Nashville, TN 37203
(615) 329-1979

Micro Computer Systems
Knoxville, TN 37922
(615) 966-9849

TEXAS

Altair Computer Center
Houston, TX 77036
(713) 780-8981

The Computer Shop
Vanguard Systems
San Antonio, TX 78216

Electrotex
Houston, TX 77006
(713) 526-3456

Interactive Computers
Houston, TX 77036
(713) 781-2703

K. A. Electronics
Dallas, TX 75247
(214) 634 7870

Micro Store
Richardson, TX 75080
(214) 231-1096

Southwest Technical Products
San Antonio, TX 78216

UTAH

Byte Shop/Salt Lake City
Salt Lake City, UT 84111

VIRGINIA

Computer Hobbies Unl.
Richmond, VA 23235
(804) 276-5056

Media Reactions Inc.
Reston, VA 22090
(703) 471-9330

Shire Enterprises
Richmond, VA 23222
(804) 321-4560

WASHINGTON

Almac/Strom Elec.
Seattle, WA 98108
(206) 763-2300

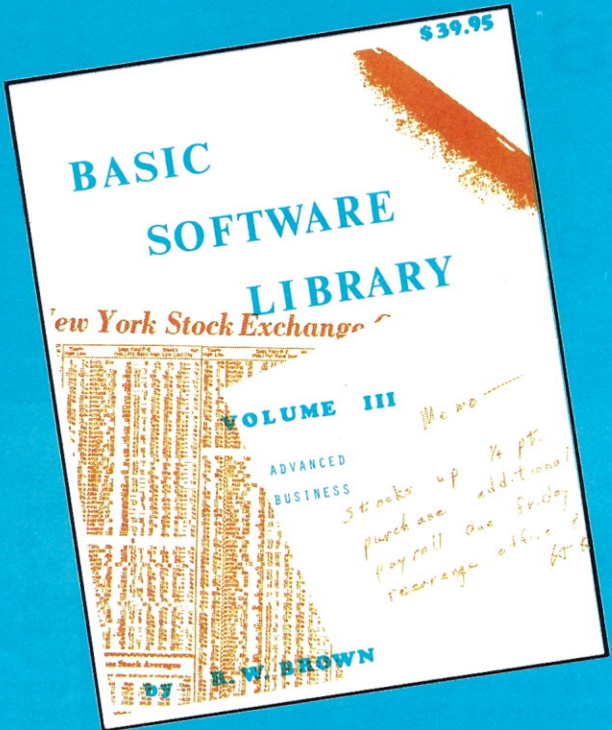
WISCONSIN

Microcomp
Fond du Lac, WI 54935
(414) 922-2515

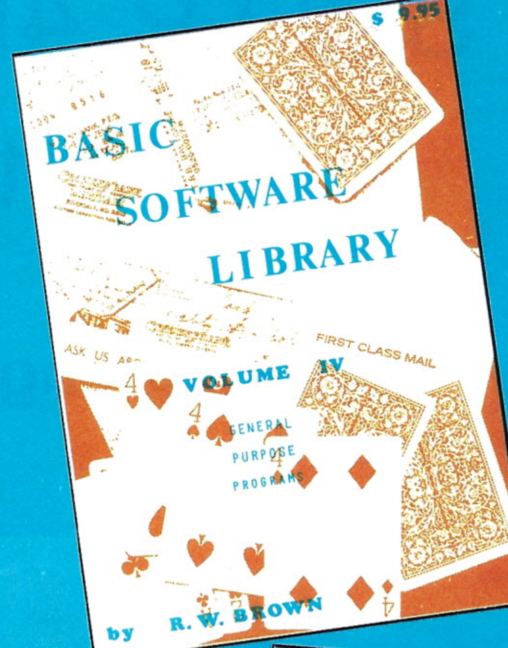
The Milwaukee Computer Store
Milwaukee, WI 53213
(414) 259-9140

WARE LIBRARY

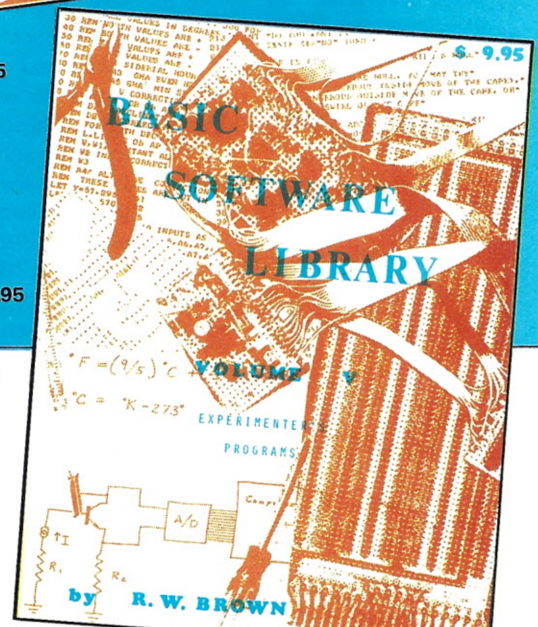
This LIBRARY is a complete do it yourself kit.
Knowledge of programming not required.
EASY to read and USE.



VOLUME III ADVANCED BUSINESS
BILLING
INVENTORY . . $39.95
PAYROLL



VOLUME IV
GENERAL PURPOSE . . $9.95



VOLUME V
EXPERIMENTER . . $9.95

This library is the most comprehensive work of its kind to date. There are other software books on the market but they are dedicated to computer games. The intention of this work is to allow the average individual the capability to easily perform useful and productive tasks with a computer. All of the program contained within this Library have been thoroughly tested and executed on several systems. Included with each program is a description of the program, a list of potential users, instructions for execution and possible limitations that may arise when running it on various systems. Listed in the limitation section is the amount of memory that is required to store and execute the program.

Each program's source code is listed in full detail. These source code listings are not reduced in size but are shown full size for increased readability. Almost every program is self instructing and prompts the user with all required running data. Immediately following the source code listing for most of the programs is a sample executed run of the program.

The entire Library is 1100 pages long, chocked full of program source code, instructions, conversions, memory requirements, examples and much more. ALL are written in compatible BASIC executable in 4K MITS, SPHERE, IMS, SWTPC, PDP, etc. BASIC compilers available for 8080 and 6800 under $10 elsewhere.

This Library is destined to become one of the reference bibles for the small computer field, due to its versatility and uniqueness and the ease of operation of the programs it contains. These volumes are deductible as a business expense when purchased by a company. Send your remittance for prompt delivery, while supplies last. Volume discounts are available to qualified dealers.

FUTURE ADDITION TO THE "BASIC SOFTWARE LIBRARY"
Volume VI (A Complete Business System — $49.95) General Ledger System — Taxes, Pyrl, W-2's, Inventory, Depr., Financial Statements, etc. AVAILABLE MID SUMMER

SCIENTIFIC RESEARCH

SRI

master charge THE INTERBANK CARD

1712-K FARMINGTON COURT
CROFTON MD 21114
Phone Orders call (800) 638-9194
Information and Maryland Residents Call (301)-721-1148

Your BANKAMERICARD welcome here

Add $1.50 per volume for postage and handling.

SWTP 4K BASIC Notes

. . . implementing it on the 680b

Stuart Mitchell
14761 Dodson Dr.
Woodbridge VA 22193

Phil Poole
1408 Idaho
Woodbridge VA 22191

4K BASIC for the Mits 680b sounded like a good place to head after we exhausted the things we could do with Mr. Pittman's Tiny BASIC. Having the Mits monitor and wanting to keep it intact was a deciding factor in our approach, especially since it is easier to use than MIKBUG. We considered, for about one microsecond, the easy approach of buying Mits BASIC for $200 before looking elsewhere. In the third issue of *Kilobaud*, you should have noted a less expensive memory system for the 680b, and that should have told you that we operate on as lean a budget as possible. The advertisements for the Southwest Technical Products' 4K BASIC looked good. The BASIC was for a 6800 microprocessor and the price was right, so we bought it. By the time it arrived, we were really ready to go with 13K of memory in our machines. We read the tape into memory and jumped to the starting location – nothing – oh well, we should have read the instructions first. It was then that we learned about MIKBUG and its temporary storage starting at location A000. After a little book work things began to fall into place and 4K BASIC is alive and well today in our 680bs.

Theory

The SWTPC 4K BASIC version 1.0 was written by Mr. Robert Uiterwyk. Since it was intended to be used with the SWTPC 6800 machine, the input and outputs were to operate with the MIKBUG monitor which Motorola Semiconductor had designed. The best we could tell from inspection was that there were only four patches required to the 4K BASIC interpreter, three outputs and one input. We do not have MIKBUG in our machine, so a software linkage of some sort is required.

Where do you put a linkage program? A quick review shows that locations 0000 to 0100 are reserved for the Mits monitor. The 4K BASIC from 0100 to 11FF, MIKBUG and 4K BASIC both use the locations from A000 to A0FF for temporary storage, and the Mits monitor is located between FF00 and FFFF. Well, our machines had memory from 0000 to 33FF and the monitor was located in FF00 to FFFF. We reconfigured the original 1K memory which was at 3000 to 33FF and placed it in locations A000 to A3FF. This can be accomplished by changing the jumpers on the main 680b printed circuit boards as instructed by Mits. You may want to install switches if you experiment as much as we do, as then you can assign the original 1K memory anywhere easily.

The table provides you with two choices for locating the software linkage, either RAM or EROM. We suggest the RAM approach to start with and then at a latter date burn it into a 1702A if you like. The 1702A will hold lots more than this linkage program. We currently have the 4K BASIC linkage, a Tiny BASIC linkage, a memory test, a dump and a move program in 1702As. Use the locations A100-A13A for the linkage program. The linkage may be moved anywhere as it is all relative with the exception of location A128 which must be in the extended mode and point to the first OUTEEE location. We have attempted to use the MIKBUG nomenclature wherever possible to make references as easy as possible,

but no claims are made by the authors.

Implementation

As for the how to do it, we suggest the following order. With the machine up and operating with 8K in low memory, the original 1K memory at location A000 and the Mits monitor at FF00, load the linkage from location A100 to A13A. Then load the 4K BASIC and inspect and compare locations 025F through 026C with the original instructions in the table. Replace the original instructions with the ones listed in the RAM column. Using the Mits monitor J command, type J 0100. The 680b should respond with the word READY. We suggest you study the instructions starting in Appendix E of the manual you bought with the 4K BASIC for a complete description. To re-enter BASIC without destroying everything requires you to jump to 0103 instead of 0100. We point this out because if you type J 0100 you'll loose the whole program you have been typing in BASIC. The # sign after each entry says you are in BASIC, and the . says you are back in the Mits monitor.

The table also lists the locations and instructions for a 1702A EROM. Use of the EROMs saves loading the linkage program at locations A100-A138 each time. But you still must put the patches at 025F to 026C and the machine must still be configured with the 1K memory at location A000 for the temporary stack storage. It is a good idea to make a new tape of the 4K BASIC with the patches inserted to guard against accidents.

Conclusions

We have had this version of 4K BASIC up and operating on our machines for 3 months now and are reasonably sure all is well. At first we used the RAM procedure but both of us have 1702A EROMs and a magnetic tape with the patches now as it is more convenient. There may be a less expensive solution. If you have one, let us know. We would also be interested in any comments on additions to the 4K BASIC, like trig functions or format statements. ■

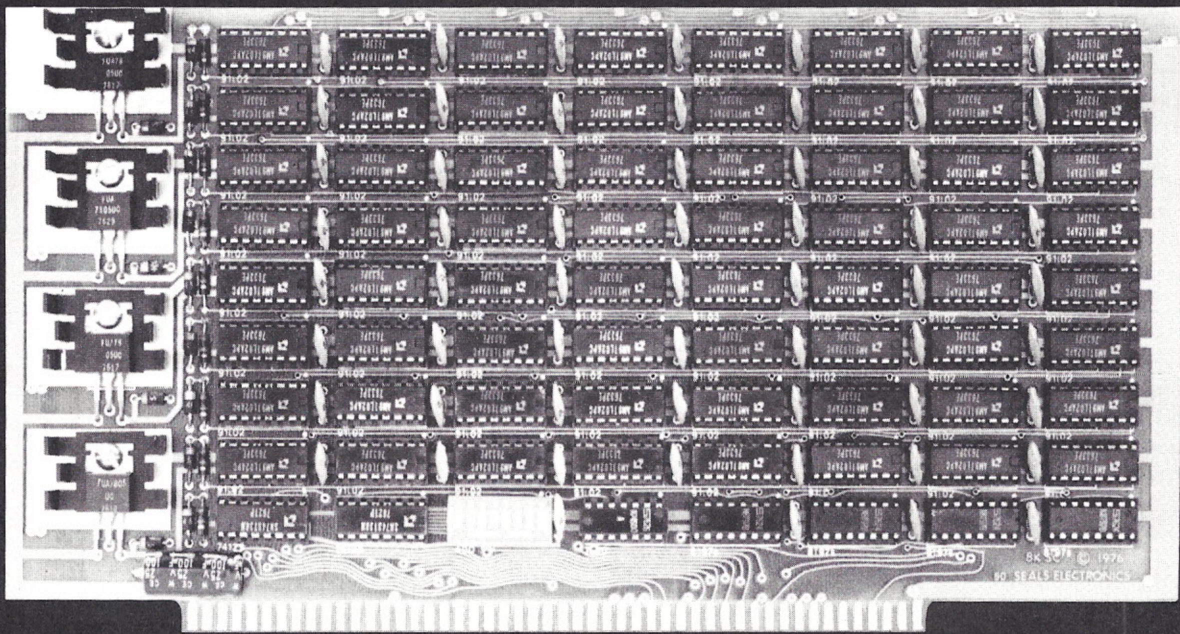
Patches to SWTPC 4K BASIC Version 1.0

Location	Instruction			Operation
	Original	RAM	EROM	
025F	7EE0BF	7EA128	7EFE2B	OUT 2HEXCH+SP
0262	7EE0C8	7EA134	7EFE34	OUT 4HEXCH+SP
0267	7EE1D1	7EA100	7EFE00	OUTEEE
026A	BDE1AC	7EA111	7EFE11	INEEE

Additional Subroutines to Provide Link to Mits Monitor

Location			
RAM	EROM	Instruction	Operation
OUTEEE			
A100	FE00	819A	CMPA
A102	FE02	2F0C	BLE
A104	FE04	36	PUSHA
A105	FE05	B6F000	LDAA
A108	FE08	8402	ANDA
A10A	FE0A	27F9	BEO
A10C	FE0C	32	PULA
A10D	FE0D	B7F001	STAA
A1100	FE10	39	RTS
INEE			
A111	FE11	37	PSHB
A112	FE12	BDFF00	INCH MITS
A115	FE13	17	TBA
A116	FE16	33	PULB
A117	FE17	7E026D	JUMP TO 4K BASIC
OUTHL			
A11A	FE1A	44	LSRA
A11B	FE1B	44	LSRA
A11C	FE1C	44	LSRA
A11D	FE1D	44	LSRA
OUTHR			
A11E	FE1E	840F	AND A
A120	FE10	8B30	ADD A
A122	FE22	8139	CMPA
A124	FE24	2302	BLS OUTCH
A126	FE26	8B07	ADD A
OUTCH			
A128		7EA100	JUMP TO OUTEEE (RAM)
	FF28	7EFE00	JUMP TO OUTEEE (EROM)
OUT2H			
A12B	FE2B	A600	LDA
A12D	FE2D	8DEB	BSR TO OUTHL
A12F	FE2F	A600	LDA
A131	FE31	08	INX
A132	FE32	20EA	BRA TO OUTHR
OUTH4HCS			
A134	FE34	8DF5	BSR TO OUT2H
OUT2HCS			
A136	FE36	8DF3	BSR TO OUT2H
OUTS			
A138	FE38	8620	LDA
A13A	FE3A	20EC	BRA OUTCH

Program Listing.



8KSC-Z

(Static Memory Card)

SPECIFICATIONS

- ALTAIR,® IMSAI,® and S-100 buss compatible.
- Access Time: 250 nsec max.
- Memory Chip: 2102LHPC or 2102AL-2
- Zilog Speed: Compatible up to 4 Mhz
- Battery Standby: 1.5 to 4V
- Address Select: 8 ea. SPST Dip Switch
- Wait States: None
- Current Reg.: Less than 200 ma per 1K
- All Address, Control and Data Out lines fully buffered

FEATURES

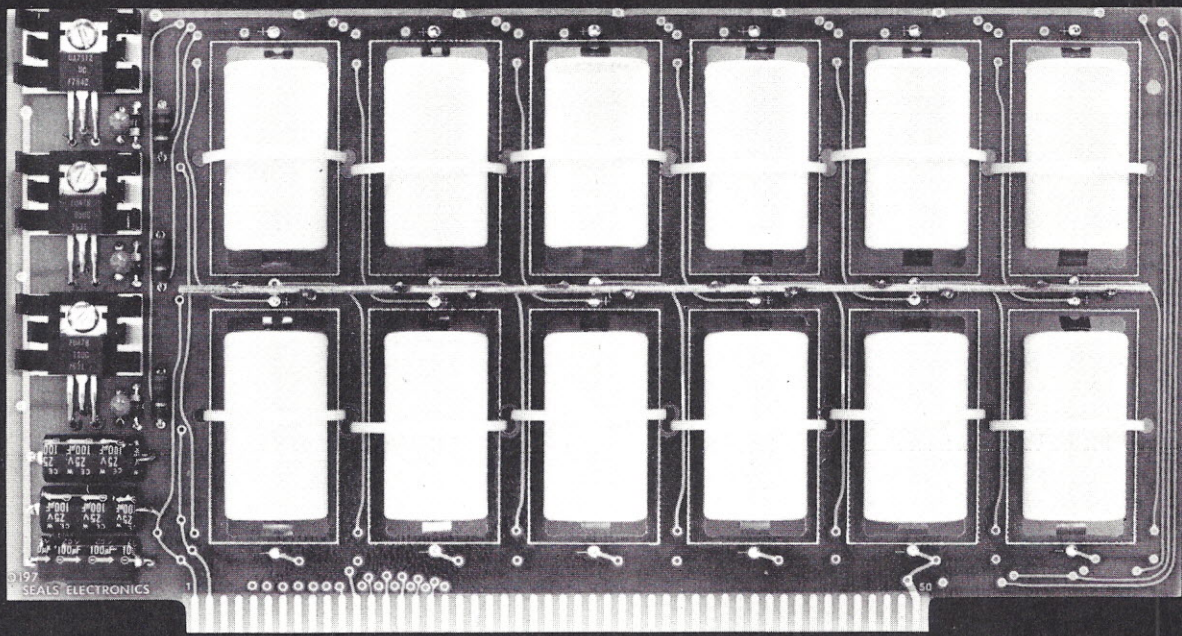
- ONLY QUALITY PARTS USED TO GIVE YOU PREMIUM QUALITY PRODUCTS
- MEMORY CHIPS MEET MILITARY SPECS
- HEAVY G-10 EPOXY GLASS PC BOARD MATERIAL WITH 2 OZ. COPPER CLAD
- COMPONENT LEGENDS SILK SCREENED ON COMPONENT SIDE OF BOARD
- HEAVY GOLD PLATED EDGE CONTACTS
- NO JUMPERS ON PC BOARDS
- ALL OPTIONS ON PC BOARDS ARE DIP SWITCH SELECTABLE
- SOLDER MASK ON BOTH SIDES OF PC BOARD
- IC SOCKETS WITH EVERY IC
- COMPLETE GUARANTEE ON ANY SEALS MANUFACTURED COMPONENT

ASSEMBLY AND OPERATING MANUAL AVAILABLE

SEALS ELECTRONICS, INC.®
BOX 11651, KNOXVILLE, TN. 37919

TELEX NO. 55-7444

TELEPHONE NO. 615/693-8655



BBUC

(Battery Backup Card)

SPECIFICATIONS

- Automatic Battery Charging Circuit
- Selectable Standby Voltage Outputs
- Will hold up to 12 ea. "C" Cell Ni-Cad batteries
- The BBUC comes selected for 2.5 Volts Standby to Pin no. 14 on the ALTAIR® or S-100 buss structure to power up our 8KSC-Z memory boards
- Can be wired to backup any Memory Card which has Battery Standby Capabilities
- Eliminates cluge wires on top of Memory Boards. (Utilizes any vacant buss lines.)
- Just plug the BBUC into any available buss connectors

FEATURES

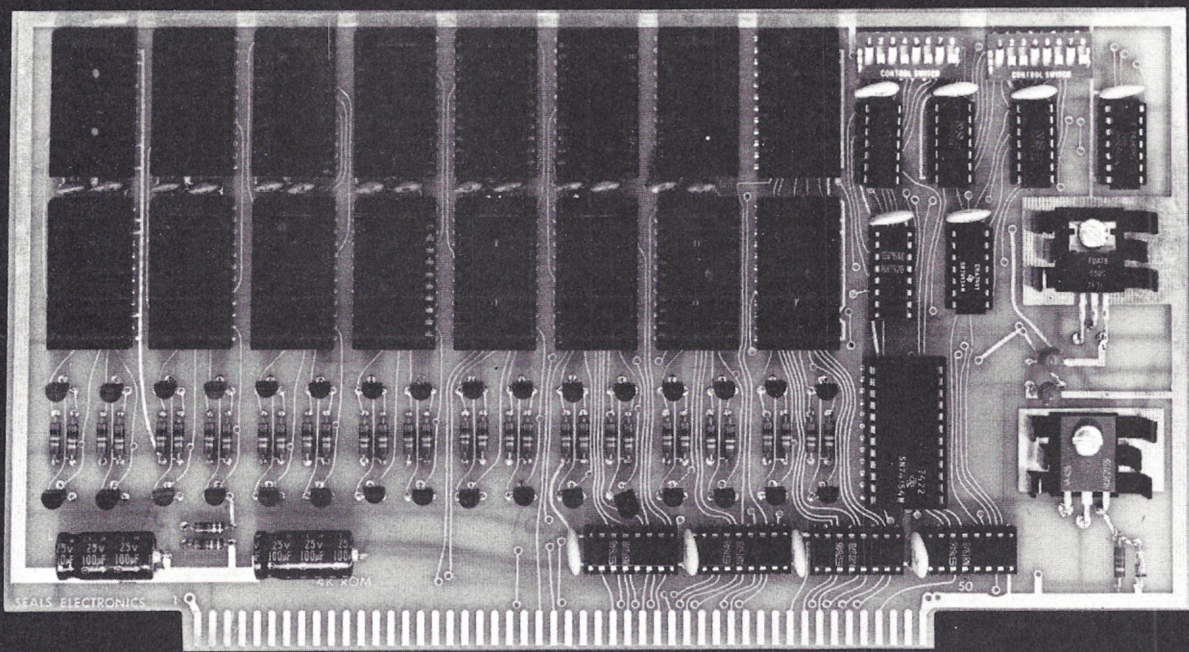
- ONLY QUALITY PARTS USED TO GIVE YOU PREMIUM QUALITY PRODUCTS
- HEAVY G-10 EPOXY GLASS PC BOARD MATERIAL WITH 2 OZ. COPPER CLAD
- COMPONENT LEGENDS SILK SCREENED ON COMPONENT SIDE OF BOARD
- HEAVY GOLD PLATED EDGE CONTACTS
- NO JUMPERS ON PC BOARDS
- SOLDER MASK ON BOTH SIDES OF PC BOARD
- IC SOCKETS WITH EVERY IC
- COMPLETE GUARANTEE ON ANY SEALS MANUFACTURED COMPONENT

ASSEMBLY AND OPERATING MANUAL AVAILABLE

SEALS®
ELECTRONICS, INC.
BOX 11651, KNOXVILLE, TN. 37919

TELEX NO. 55-7444

TELEPHONE NO. 615/693-8655



4KROM

(Read Only Memory)

SPECIFICATIONS

- Will accept up to 16 ea. 1702A or 5203 EProm providing up to 4096 words of non-volatile memory for Boot Loads to Complete Programs
- Programming available at factory for $3.00 per EProm when accompanied by binary formated tape.
- Each 1702A has its own Vgg clocked for Low power consumption. Will work with the weakest power supply based S-100 buss computer.
- Switched Selected Address in 4K Blocks.
- Switch selected wait states so that even the slowest 1702A can work in your system. 0-7 wait states.

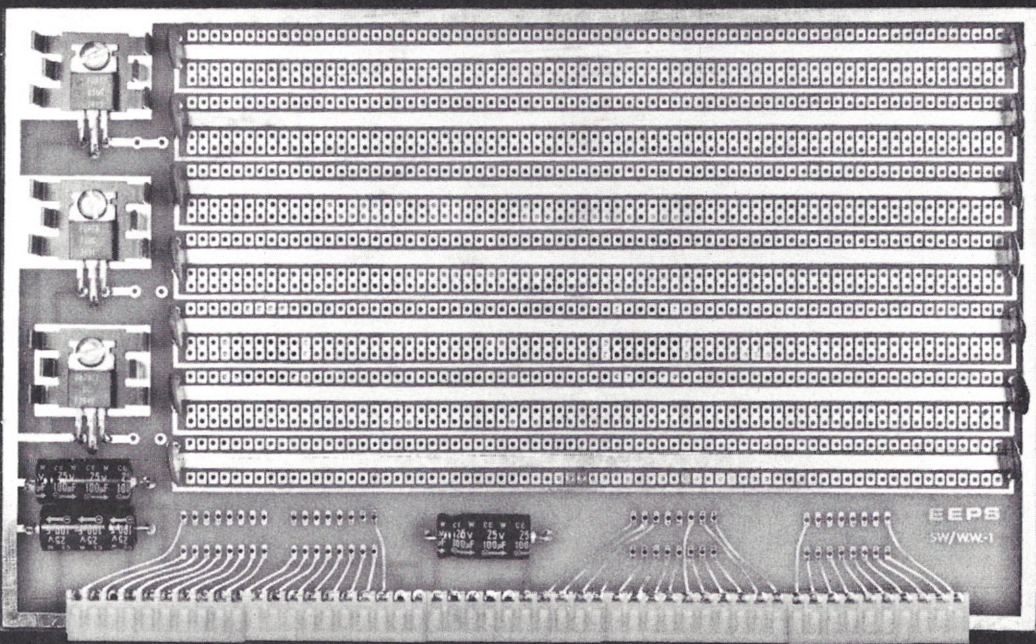
FEATURES

- ONLY QUALITY PARTS USED TO GIVE YOU PREMIUM QUALITY PRODUCTS
- MEMORY CHIPS MEET MILITARY SPECS
- HEAVY G-10 EPOXY GLASS PC BOARD MATERIAL WITH 2 OZ. COPPER CLAD
- COMPONENT LEGENDS SILK SCREENED ON COMPONENT SIDE OF BOARD
- HEAVY GOLD PLATED EDGE CONTACTS
- NO JUMPERS ON PC BOARDS
- ALL OPTIONS ON PC BOARDS ARE DIP SWITCH SELECTABLE
- SOLDER MASK ON BOTH SIDES OF PC BOARD
- IC SOCKETS WITH EVERY IC
- COMPLETE GUARANTEE ON ANY SEALS MANUFACTURED COMPONENT

ASSEMBLY AND OPERATING MANUAL AVAILABLE

SEALS® ELECTRONICS, INC.
BOX 11651, KNOXVILLE, TN. 37919

TELEX NO. 55-7444
TELEPHONE NO. 615/693-8655



68WWC (Wire Wrap Card for SWTPC 6800)

SPECIFICATIONS

- Contains all Buss Connectors
- Contains 3 ea. Voltage Reg. +12, -12, +5
- Contains 3 ea. 100 uf 35 Volt input Cap.
- Contains a generous supply of .1 12V decoupling capacitors
- Will accept all popular IC wire wrap sockets 40, 24, 22, 16, 14, etc.
- Highest quality proto-typing board on the market today. It has all the basics. You just add your circuit
- COMPONENT LEGENDS SILK SCREENED ON COMPONENT SIDE OF BOARD
- HEAVY GOLD PLATED EDGE CONTACTS
- NO JUMPERS ON PC BOARDS
- HEAVY G-10 EPOXY GLASS PC BOARD MATERIAL WITH 2 OZ. COPPER CLAD
- SOLDER MASK ON BOTH SIDES OF PC BOARD
- IC SOCKETS WITH EVERY IC

ASSEMBLY AND OPERATING MANUAL AVAILABLE

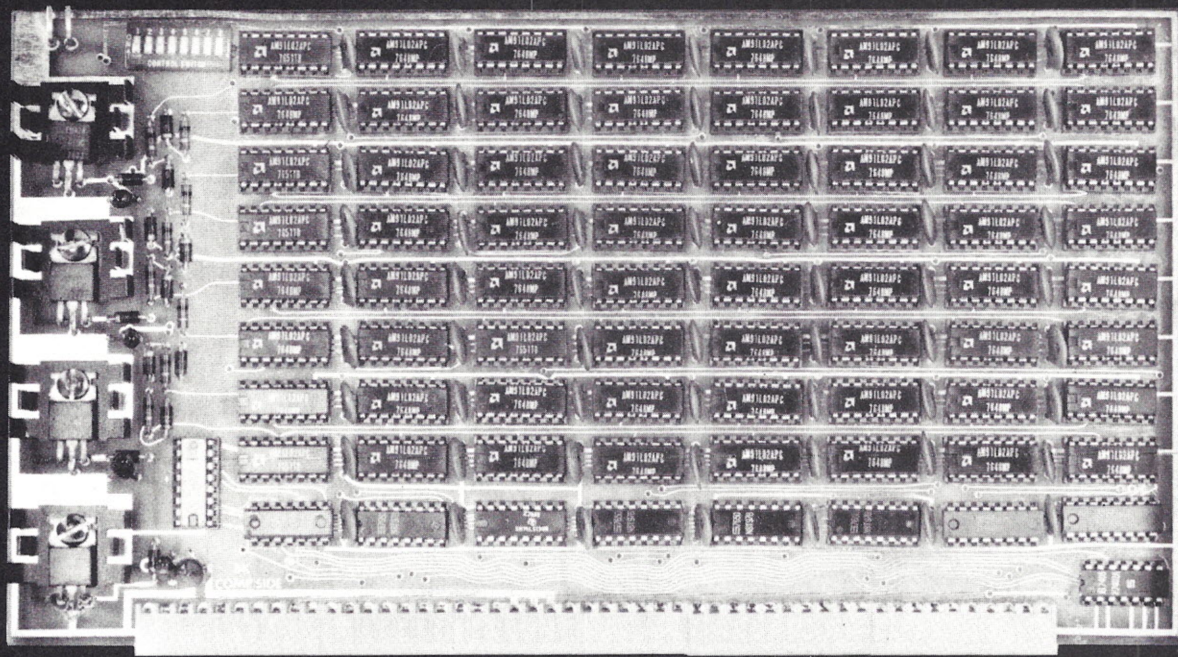
FEATURES

- ONLY QUALITY PARTS USED TO GIVE YOU PREMIUM QUALITY PRODUCTS
- COMPLETE GUARANTEE ON ANY SEALS MANUFACTURED COMPONENT

SEALS ®
ELECTRONICS, INC.
BOX 11651, KNOXVILLE, TN. 37919

TELEX NO. 55-7444

TELEPHONE NO. 615/693-8655



68KSC

(Static Memory Card)

SPECIFICATIONS

- All lines fully buffered
- 8192 words of Static Memory
- Access Time: 500 nsec. (250 nsec on request)
- Memory Chip: 91LO2 APC or 2102AL-4
- Battery Standby
- Address Selected 8 ea. SPST Dip Switch
- Low Power

- COMPONENT LEGENDS SILK SCREENED ON COMPONENT SIDE OF BOARD
- HEAVY GOLD PLATED EDGE CONTACTS
- NO JUMPERS ON PC BOARDS
- ALL OPTIONS ON PC BOARDS ARE DIP SWITCH SELECTABLE
- SOLDER MASK ON BOTH SIDES OF PC BOARD
- IC SOCKETS WITH EVERY IC
- COMPLETE GUARANTEE ON ANY SEALS MANUFACTURED COMPONENT
- DMA SPEEDS TO 2 Mhz

ASSEMBLY AND OPERATING MANUAL AVAILABLE

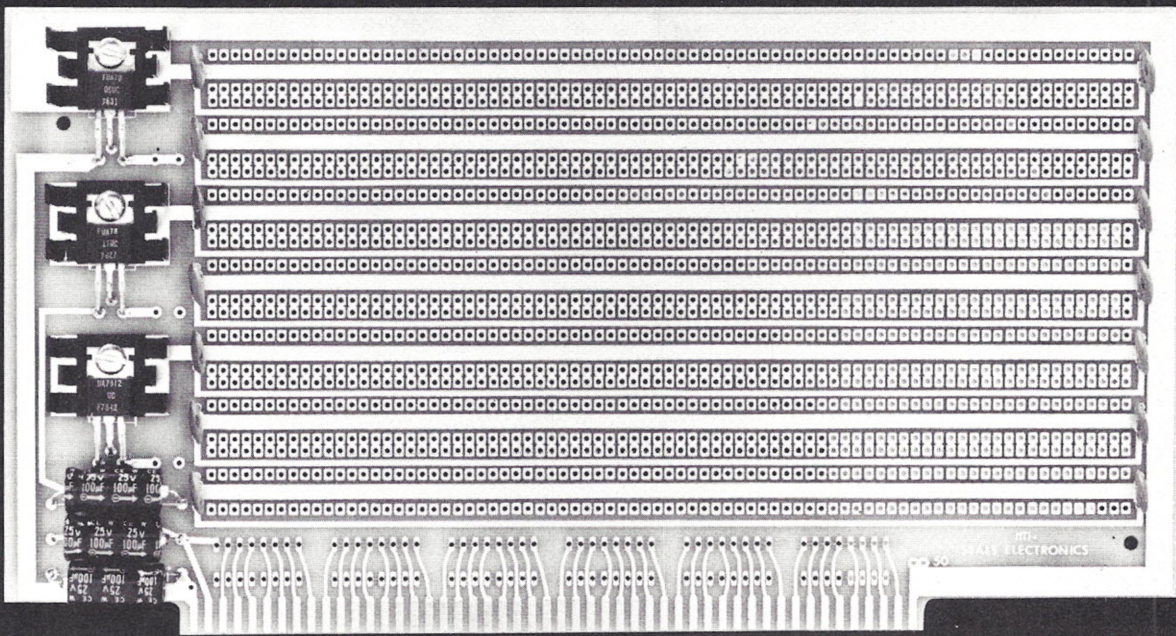
FEATURES

- ONLY QUALITY PARTS USED TO GIVE YOU PREMIUM QUALITY PRODUCTS
- MEMORY CHIPS MEET MILITARY SPECS
- HEAVY G-10 EPOXY GLASS PC BOARD MATERIAL WITH 2 OZ. COPPER CLAD

SEALS ®
ELECTRONICS, INC.
BOX 11651, KNOXVILLE, TN. 37919

TELEX NO. 55-7444

TELEPHONE NO. 615/693-8655



88WWC Wire Wrap Card

SPECIFICATIONS

- Accepts all IC sockets
- 3 Voltage Regulators +12, -12, +5
- Contains 3 ea. 100 uf 35 Volt input Cap.
- Contains a generous supply of .1 12V decoupling capacitors

FEATURES

- ONLY QUALITY PARTS USED TO GIVE YOU PREMIUM QUALITY PRODUCTS
- HEAVY G-10 EPOXY GLASS PC BOARD MATERIAL WITH 2 OZ. COPPER CLAD
- COMPONENT LEGENDS SILK SCREENED ON COMPONENT SIDE OF BOARD
- HEAVY PLATED THROUGH HOLES
- HEAVY GOLD PLATED EDGE CONTACTS
- NO JUMPERS ON PC BOARDS
- SOLDER MASK ON BOTH SIDES OF PC BOARD
- IC SOCKETS WITH EVERY IC
- COMPLETE GUARANTEE ON ANY SEALS MANUFACTURED COMPONENT

ASSEMBLY AND OPERATING MANUAL AVAILABLE

SEALS ®
ELECTRONICS, INC.
BOX 11651, KNOXVILLE, TN. 37919

TELEX NO. 55-7444

TELEPHONE NO. 615/693-8655

SEALS DEALERS/DISTRIBUTORS*:

ALABAMA

JACK'S T.V. SUPPLY, INC.
Box 10482
Birmingham, Ala. 35202
(205) 328-9890

ARIZONA

BYTE SHOP
813 N. Scottsdale Road
P.O. Box 28106
Tempe, Arizona 85281
(602) 894-1129

ARIZONA MICRO SYSTEMS
3240 West Larkspur
Phoenix, Arizona 85029
(602) 942-8405

CALIFORNIA

CHANNEL RADIO AND ELECTRONICS
18 East Ortega Street
Santa Barbara, CA. 93101
(804) 965-8551

COMPUTER STORE of SAN FRANCISCO
1093 Mission Street
San Francisco, CA 94103
(415) 431-0640

THE COMPUTER MART
625 W. Katella Avenue No. 10
Orange, CA 92667
(714) 633-1222

BYTE SHOP of TARZANA
18424 Ventura Blvd.
Tarzana, CA 91356
(213) 343-3421

BYTE SHOP of LAWNDALE
16508 Hawthorne Blvd.
Lawndale, CA. 90260
(213) 371-3421

BITS, BYTES AND PIECES
6211 Quincewood Circle
Citrus Heights, CA. 95610
(916) 726-6469

***COMPUTER MART DISTRIBUTING CO.**
625 W. Katella
Orange, CA. 92667
(714) 633-4634

A-VIDD ELECTRONICS
2210 Bellflower Blvd.
Long Beach, CA. 90806
(213) 426-5526

BYTE SHOP of HAYWARD
1122 "B" Street
Hayward, CA. 94541
(415) 537-BYTE

BYTE SHOP COMPUTER STORE
4 W. Mission
Santa Barbara, CA. 93101
(805) 966-2638

BYTE SHOP of LONG BEACH
5453 E. Stearns Street
Long Beach, CA. 90815
(213) 5977771

BYTE SHOP of PASADENA
496 South Lake Avenue
Pasadena, CA 91101

BERKELEY BYTE SHOP
1514 University Avenue
Berkeley, CA. 94703

BYTE SHOP of SACREMENTO
6041 Greenback Lane
Citrus Heights, CA 95610
(916) 726-2557

COMPUTER COMPONENTS
5848 Sepulveda Blvd.
Van Nuys, CA 91411

THE COMPUTER CENTER
8205 Ronson Road
San Diego, CA 92111
(714) 292-5302

COMPUTER ELECTRONICS
905 De La Vina
Santa Barbara, CA 93101
(805) 962-4198

BYTE SHOP of SAN JOSE
2626 Union Avenue
San Jose, CA 95124

ALPHA MICROSYSTEMS COMPANY
Suite n, 17875 Sky Park North
Irvine, CA 92714
(714) 957-1404

ALLIED COMPUTER STORES
3152 Bermuda Drive
Costa Mesa, CA 92626

ALLIED COMPUTER STORES, INC.
1200 W. Hillsdale Blvd.
San Mateo, CA 94403
(415) 341-4200

AAAA COMPUTER HOW'S
1477 Barrington no. 17
West Los Angeles, CA 90025

COMPUTER DESIGNS
4943 Bosworth Court
Newark, CA 94560
(415) 792-1376

ORANGE COUNTY COMPUTER CENTER
1913 Harbor Blvd.
Costa Mesa, CA 92627

SUNSHINE COMPUTER COMPANY
20710 South Leapwood Avenue
Carson, CA 90746
(213) 327-2118

TECH-MART
19590 Ventura Blvd
Tarzana, CA 91356
(213) 344-0153

XIMEDIA CORPORATION
1290 24th Avenue
San Francisco, CA 94122
(800) 227-4440

CANADA

FOCUS SCIENTIFIC
160 Elgin Street
Ottawa, Canada K2P 204
(613) 236-7767

THE COMPUTER PLACE
186 Queen Street West
Toronto, Canada M5V 121
(416) 598-0260

***TRINTRONICS LIMITED**
186 Queen Street, W.
Toronto, Ontario, Canada M5V 1Z1
(416) 598-0262

CENTRAL-DYNAMICS
147 Hyms Blvd.
Pointe Claire, Quebec
CANADA H94 1G1

COLORADO

BYTE SHOP/ARAPAHOE CO.
3464 S. Acoma
Englewood, CO. 80110
(303) 761-6232

DIAL ELECTRIC & ENGINEERING
7121 Julian Street
Westminster, CO 80030

THE COMPUTER HUT
1764 Blake Street
Denver, CO 80202
(303) 573-4895

THE BYTE SHOP
2040 30th Street, Suite B-2
Boulder, CO 80301

CONNECTICUT

NATIONAL COMMUNICATION INDUSTRIES CO.
One River Road
Cos Cob, Conn. 06807
(203) 661-2800

FLORIDA

SUNNY COMPUTER STORES, INC.
University Shopping Center
1238 A.S. Dixie Highway
Coral Gables, Fla. 33146
(305) 661-6042

WILLIAMS RADIO TELEVISION, INC.
2062 Liberty Street
P.O. Box 3314
Jacksonville, Fla. 32206
(904) 354-5460

COMPUTER HUT
5905 N.W. 151st Street
Miami Lakes, Fla. 33014
(305) 821-2667

BYTE SHOP of MIAMI
7825 Bird Road
Miami, Fla. 33155
(305) 264-BYTE

DATA TECHNOLOGY ASSOCIATE CORPORATION
P.O. Box 1912
Miami, FA 33143

MARSH DATA SYSTEMS
5405 B Southern Comfort Blvd.
Tampa, FA 33614
(813) 886-9890

BYTE SHOP COMPUTER STORE
P.O. Box 443
1325 North Atlantic Ave.
Suite 4
Cocoa Beach, FA 32931
(305) 784-1881

GEORGIA

ATLANTA COMPUTER MART
5091-B Buford Highway
Atlanta, GA. 30340
(404) 455-0647

INDIANA

C & I COMMUNICATION
P.O. Box 52
Interstate 70, State no. 1
Cambridge, Inc. 47327
(317) 478-3749

QUANTUM COMPUTER WORKS
6637 Kennedy Ave.
Hammond, Ind. 46323
(219) 989-9828

DATA DOMAIN of INDIANAPOLIS
7027 Michigan Road
Indianapolis, Ind. 46268
(317)251-3139

The DATA DOMAIN of WEST LAFAYETTE INDIANA
219 Columbia Ave.
West LaFayette, Inc. 47908

***HOBBYTRONIC DISTRIBUTORS**
1218 Prairie Drive
Bloomington, Ind. 47401
(812) 336-6380

THE DATA DOMAIN
506 South College
Bloomington, Ind. 47401
(812) 334-3607

THE DATA GROUP, INC. BYTE SHOP
5947 East 82nd Street
Indianapolis, Ind. 46250
(317) 842-2983

HOME COMPUTER CENTER
2115 E. 62nd Street
Indianapolis, Ind. 46220
(317) 251-6800

ILLINOIS

itty bitty MACHINE COMPANY of EVANSTON
1316 Chicago Ave.
Evanston, Ill. 60201
(312) 328-6800

itty bitty MACHINE COMPANY
42 W. Roosevelt
Lombard, Ill. 60148

AMERICAN MICRO-PROCESSOR EQUIPMENT & SUPPLY CORPORATION
P.O. Box 515
20 N. Milwaukee Ave.
Prairie View, Ill. 60069
(312) 634-0076

THE SYSTEMS GROUP
1327 W. 29th Street
Molin, Ill. 61265

IOWA

THE COMPUTER STORE of DAVENPORT, INC.
616 W. 35th Street
Davenport, Iowa 52806
(319)386-3334

KENTUCKY

DATA DOMAIN of LOUISVILLE
3028 Humsinger Lane
Louisville, KY 40220
(502) 456-5242

DATA DOMAIN of LEXINGTON
506 ½ Euclid Avenue
Lexington, KY
(606) 233-3346

CYBERTRONICS
312 Production Court
Louisville, KY 40299
(502) 499-1551

MARYLAND

COMPUTER WORKSHOP
1776 Plaza
1776 East Jefferson
Rockville, MD 20852
(301) 468-0455

CHEAP, INC.
7338 Baltimore Ave., Suite 200
College Park, MD 20740
(301) 779-7997

SCIENTIFIC RESEARCH INSTRUMENTS CO., INC.
1712 Farmington Court
Crofton, MD 21114
(301) 721-1148

MASSACHUSETTS

THE COMPUTER MART
1097 Lexington
Waltham, MA 02154
(617) 899-4540

MICHIGAN

NEWMAN COMPUTER EXCHANGE
1250 N. Main Street
Ann Arbor, MI 48104
(313) 994-3200

MISSOURI

COMPUTER WORKSHOP of KANSAS CITY
6903 Blair Road
Kansas City, MO 64152
(816) 741-5055

COMPUTER SYSTEMS CENTER of ST. LOUIS
13461 Olive
Chesterfield, MO 63017
(314) 576-5020

BYTE SHOP of EAGAN
7547 Irish Avenue, Court South
Cottage Grove, MO 55016

NEBRASKA

OMAHA COMPUTER STORE
4540 South 84th Street
Omaha, Nebr. 68127
(402) 592-3590

NEW JERSEY

THE COMPUTER MART of NEW JERSEY, INC.
501 Route no. 27
Iselin, NJ 08830
(201) 283-0600

WILLIAM ELECTRONIC SUPPLY
1863 Woodbridge Ave.
Edison, NJ 08817
(201) 985-3700

NEW YORK

THE MEMORY MERCHANT
P.O. Box "O"
Spencerport, NY 14559
(716) 352-1325

THE COMPUTER MART
118 Madison
New York, NY 10016
(212) 686-7923

***MJB RESEARCH AND DEVELOPMENT**
36 W. 62nd Street
New York, NY 10023
(212) 245-8530

COMPUTER MART of LONG ISLAND
East Meadow, NY 11554
(516) 794-0510

COMPUTER ENTERPRISES
3343 Erie Blvd., E.
Syracuse, NY 13214

OHIO

CY-COMP
1154 Desert Street
Uniontown, Ohio 44685
(216) 877-6423

DIGITAL DESIGN
7694 Camargo Road
Cincinnati, Ohio 45243
(513) 561-6733

ELS SYSTEMS
2209 N. Taylor Road
Cleveland, Ohio 44112
(216) 249-7820

OREGON

BYTE SHOP, INC.
2033 S.W. 4th
Portland, OR 97201
(503) 223-3496

SOUTH CAROLINA

BYTE SHOP of SOUTH CAROLINA
2018 Green Street
Columbia, SC 29207
(803) 771-7824

TENNESSEE

MICRO COMPUTER SYSTEMS
Route 36, McFee Road
Knoxville, TN 37922
(615) 966-9849

***MCED COMPANY**
Suite 101
1600 Hayes Street
Nashville, TN 37203
(615) 329-1979

COMPUTER DENN
1507-A Oak Ridge Turnpike
Oak Ridge, TN 37830
(615) 482-1091

TEXAS

BYTE SHOP COMPUTER STORE
3211 Fondren
Houston, Tex. 77063
(713) 977-0664

TANNER ELECTRONICS
11423 Harry Hines
Dallas, Tex. 75229

MICROTEX, INC.
9305-D Harwin Drive
Houston, Tex. 77036

COMPUTER SHOPS, INC.
13933 N. Central Expressway
Suite 211
Dallas, Tex. 75243

PENNSYLVANIA

PERSONAL COMPUTER CORPORATION
Routes 30 and 352
Frazer Mall
Frazer, PA 19355
(215) 647-8463

UTAH

THE COMPUTER ROOM
3455 South West Temple
Salt Lake City, UT 84115
(801) 486-4311

VIRGINIA

COMPUTER SYSTEMS STORE
1984 Chainbridge Road
McLean, VA 22101
(703) 821-8333

EMR SPECIALISTS
1921 Dogwood Lane
P.O. Box 167
Vienna, VA 22180

WASHINGTON

ALMAC STROUN ELECTRONICS
5811 6th Avenue, South
Seattle, WA 98108

MICRO COMPUTER CENTER
11822 Northeast 8th
Bellevue, WA 98005
(206)455-3710

WISCONSIN

AUSTIN COMPUTERS
1835 North Gate
Beloit, WI 53511
(608) 365-6095

itty bitty MACHINE COMPANY
2221 East Capital
Shorewood, WI 53211
(414) 961-2430

MADISON COMPUTER STORE
1910 Monroe Street
Madison, WI 53711

MILWAUKEE COMPUTER STORE
6916 W. North Avenue
Milwaukee, WI 53213
(414) 259-9140

EUROPE & ASIA

***BARSADO LTD**
P.O. Box 9
Bracknell
Berkshire
ENGLAND
0344 3421

***DENOTES DISTRIBUTORS ONLY**

SEALS ELECTRONICS WESTERN REGIONAL OFFICE
Pete Bickerdike
178 Del Canto Lane
Santa Barbara, CA 93110
(805) 964-5063

PRICE LIST
FOR SEALS ELECTRONICS, INC.

ITEM NO.	DESCRIPTION	KIT PRICE	ASSEMBLED PRICE
4KROM	4K Read Only Memory Card	$119.00	$179.00
8KSC	8K Static Memory Card 500 nsec	269.00	369.00
8KSC-Z	8K Static Memory Card 250 nsec	295.00	395.00
BBUC	Battery Back-Up Card	55.00	68.00
68 KSC	8K Static Memory Card SWTPC 6800 Compatible Products	269.00	369.00
68WWC	Wire Wrap Card SWTPC 6800 Compatible Products	35.00	45.00
68 EXT-S	Extender Card [Small] SWTPC 6800 Compatible Products	19.00	25.00
68 EXT-L	Extender Card [Large] SWTPC 6800 Compatible Products	29.00	39.00
88 WWC	Wire Wrap Card	37.50	47.50
88 EXT	88 Extender Card	29.00	38.00

ASSEMBLY AND OPERATING MANUALS AVAILABLE FOR ALL ITEMS . . . $4.00 ea.

EFFECTIVE DATE JUNE 1, 1977

ORDERFORM

SC-1

NAME ____________________
PLEASE PRINT OR TYPE
ADDRESS ____________________
CITY ____________ STATE ________ ZIP ______

PAYMENT ENCLOSED

SEND CHECK • MONEY ORDER • COD'S ACCEPTED • CREDIT CARDS

☐ ☐ ☐ ☐

FOR CREDIT CARDS

COPY ACCOUNT NUMBER FROM YOUR BANK AMERICARD

BANKAMERICARD.

MY CARD EXPIRES ▶

COPY ACCOUNT NUMBER FROM YOUR MASTER CHARGE

master charge THE INTERBANK CARD

Copy number above your name on Mastercharge ▶

MY CARD EXPIRES ▶

ITEM	QTY	DESCRIPTION	PRICE
		NET AMOUNT	
		SALES TAX	
		SHIPPING & HANDLING	
		GROSS AMOUNT	

Add $2.00 per card for shipping and handling

SEND TO:

SEALS ELECTRONICS, INC.
P.O. BOX 11651
KNOXVILLE, TENNESSEE 37919
615/693-8655
TELEX NO. 55-7444

Phil Hughes
PO Box 2847
Olympia WA 98507

Hexdec

. . . hexadecimal to decimal conversion

Hexdec is a hexadecimal/decimal number conversion routine. I wrote this program because I am always trying to do these number conversions with a pencil and paper and generally the answers I get are wrong. This program converts any decimal number from 0 to 65535 to its hexadecimal equivalent and any hexadecimal number from 0 to FFFF to its decimal equivalent.

The program was written in SWTPC 8K BASIC, however it should be easy to convert to any BASIC which supports strings and substring functions. Program A is the listing and Program B shows a sample execution. ■

```
0001 REM HEXDEC 4-16-1977
0100 PRINT
0110 PRINT "THIS PROGRAM CONVERTS HEX NU
MBERS TO DECIMAL AND"
0120 PRINT "DECIMAL NUMBERS TO HEX"
0130 PRINT "IF A NUMBER IS HEXADECIMAL P
RECEED IT WITH A 'H' "
0140 PRINT
0200 DATA 4096, 256, 16, 1
0210 H$="0123456789ABCDEF"
0300 REM GET NEXT NUMBER
0310 RESTORE
0320 INPUT N$
0330 IF LEFT$(N$, 1)="H" GOTO 1000
0340 IF N$="END" THEN 2000
0400 REM DECIMAL TO HEX
0410  N=VAL(N$)
0420  X$=""
0430  J=4
0440 READ P
0450 FOR I=1 TO 16
0460 IF N-I*P <0 THEN 500
0470 NEXT I
0480 PRINT " >>>>INPUT ERROR"
0490 GOTO 300
0500  X$=X$+MID$(H$, I, 1)
0510  N=N-(I-1)*P
0520  J=J-1
0530 IF J >0 THEN 440
0540 PRINT N$; " DECIMAL IS ";X$; " HEX"
0550 GOTO 300
1000 REM HEX TO DECIMAL
1010  J=2
1020  L=LEN(N$)
1030 IF L <2 THEN 480
1040 IF L >5 THEN 480
1050 FOR I=1 TO 4
1060  O(I)=0
1070 NEXT I
1080 FOR I=6-L TO 4
1090  O(I)=ASC(MID$(N$, J,1))-48
1100 IF O(I) >9 THEN O(I)=O(I)-7
1110  J=J+1
1120 NEXT I
1130 O=4096*O(1)+256*O(2)+16*O(3)+O(4)
1140 PRINT N$; "IS";O; " DECIMAL"
1150  GOTO  300
2000  END
```

Program A.

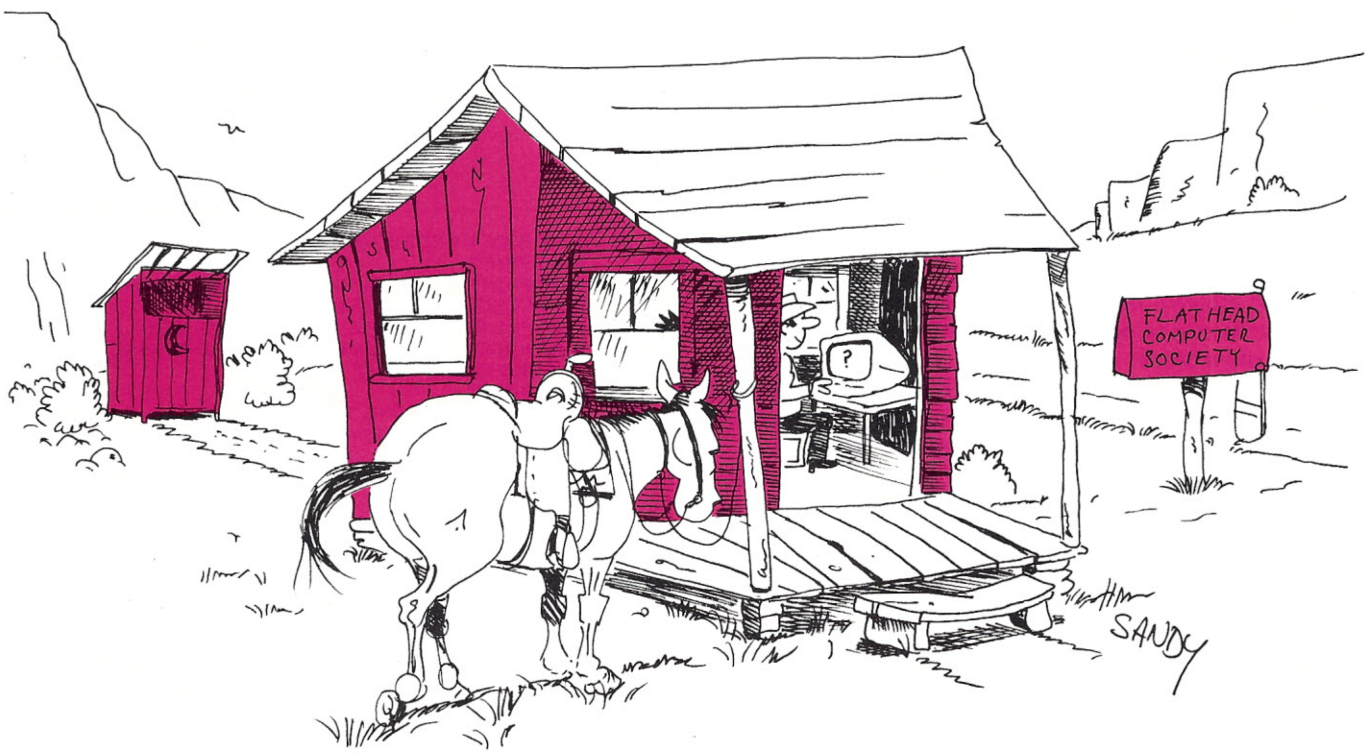
```
THIS PROGRAM CONVERTS HEX NUMBERS
TO DECIMAL AND
DECIMAL NUMBERS TO HEX
IF A NUMBER IS HEXADECIMAL PRECEED
IT WITH A 'H'

?HABCD
HABCD IS 43981 DECIMAL
?43981
43981 DECIMAL IS ABCD HEX
?H33
H33 IS 51 DECIMAL
?1234
1234 DECIMAL IS 04D2 HEX
?END

READY
#
```

Program B.

Ernie Brooner
PO Box 236
Lakeside MT 59922

Start a One-Man Computer Club

... put yourself on the map



Being a one-man computer club has both advantages and disadvantages. The main advantage is that no one can out-vote you or otherwise disrupt your schedule. The disadvantages are that there is no one to talk to, learn from, or share the joy and frustration of a new hobby. Nevertheless, it is possible to break into the hobby computer game alone until such time as someone else in your area decides to join the game.

I first heard about hobby computers six months ago, and it has been an uphill struggle ever since. Here is what I have learned so far: that many years in commercial electronics does not a computer expert make, and that a lot of youngsters who never heard of Ohm's law are way ahead of me already; that the nearest computer store is over 500 miles away; and that in spite of all that, I am learning a little bit about chips, logic, and programming. I am about halfway into my first home built computer, and if it never works at all it will have been worth the effort.

It is not easy to get into computers from somewhere in the backwoods, but it can be done. You can't find a computer store nearby? Undoubtedly then, the best way to get started is to obtain some books and magazines. Begin a collection of the free catalogs and brochures listed in the ads. Much is junk, but some is useful. Try to see and use a working hobby computer. For me that meant a 2000 mile round trip to Santa Barbara, California, a real hotbed of computer activity, where I visited with and learned from Doug Penrod and Dr. Doug Hogg (see *Kilobaud* #3 and 4).

Starting a One-man Club

If you can't find a club start one. My one-man club is listed in various national

publications and has brought me some interesting mail. For example, a quizzical letter from a novice 200 miles away who just couldn't believe such a small town had a computer club. I was also contacted by a computer store 500 miles away that put me in touch with a nearby prospective member. The newsletters of other clubs are also informative and provide contacts. When you get to this point several interested people will become visible and your days as a one-man club may be numbered.

Here is a good way to begin looking for computer information, equipment, and computer people, if you live in an isolated area. First scan the phone book. It may turn up a data processing service you don't know about or an otherwise invisible technician for one of the computer leasing companies. Check out the offices of government agencies — they are everywhere, and many use computers. Try the local library. In my case this was a blank. The most important contact in my case was a nearby community college. There I located 3 programming courses, several interested students, and an instructor who runs a part-time data processing service.

One unusual source here was the government vocational rehabilitation service. They were able to tell me the exact number of computers and programmers in the entire state. They also advised me to stay away from computers, as there was no future in them!

Publications

Excepting the college and my out-of-state computer friends, my best information source has been books and magazines. There is a bewildering choice. Take it easy on the books unless you have expert advice; many otherwise good texts are now obsolete, the contents of some cannot be identified from the titles alone, and some are a pure rip-off. The *CMOS* and *TTL cookbooks* are excellent, as is the Osborne series on microprocessor fundamentals. If your magazine store has no strictly computer mags, look for computer sections in others, such as the I/O portion of *73*.

My recommendation to a beginner is: Be prepared to spend up to $100 on publications as a starter. After wading through these you can share them with others, or drop out if you feel like it without being hurt badly. When you are ready to buy or build, here are some of the pitfalls. Hobby computer documentation is notoriously bad when compared with kit electronics in general. It is hard to get all of the parts and components you need from one or two sources, even if you can identify them, and the delay after ordering seems to be two or three times longer than if it came from the Sears catalog. If you are truly a beginner it is probably best to buy a working unit or a complete kit, but only after seeing one or talking to someone who has been through it with the same company. There are probably two dozen types of kits and several times that many options and variations.

There are certainly many places in the US, Canada, and elsewhere that as yet have no computer clubs or stores. But they do have individuals who are sincerely interested; the letters to *Kilobaud* prove this. If you are a beginner, you may be one of those individuals, and there may be another just down the block. My counsel as one of you is, don't despair. You can learn by yourself if necessary and you can ferret out that other fellow down the block. Try starting a one or two-man club . . . it won't stay that small very long in this fast-moving computer world. ■

CP-117

LETTERS

from page 13

centrate on directing anger and displeasure over such things when they're intentional. Neither the hams, with their XYL, or Tri-Tek, with their Amp'l Anny, are deliberately trying to demean or belittle women. – John.

Altair 680b Benchmark Results

Your magazine arrived today, which brightened an otherwise cloudy day. Read the article on BASIC timing comparisons. Very interesting. One thing that made Altair 680b BASIC look worse than it really is, is that the Altair 680b runs on a 500 kHz clock instead of the Motorola specification maximum of 1 MHz. The reason is that the Altair 680b uses 1702 PROMs to store the monitor program, and the system runs at half speed to accommodate the slow memory chips. So it isn't a software problem that causes the Altair 680b BASIC to run so slow, it's a hardware problem. Perhaps someday MITS, Inc. will redesign the 680b to use faster PROMs for the monitor and speed up the system clock. Actually, the Microsoft 6800 BASIC is a relatively fast program. If you were to run the benchmark programs on a system with a 1 MHz clock, the Altair 680b BASIC, written by Microsoft, would be near the top of the timing chart. Someone should run the benchmark programs on OSI 6502 BASIC. Judging from their ads, the programs should run like greased lightning.

Since you've been publishing business applications programs written in BASIC, someone should point out that not all microcomputer BASICs are suitable for running accounting programs. Most BASICs use a 4 byte binary floating point number format, which gives only 6 decimal digits. That is simply not enough for accounting. In addition, errors can creep in the conversion from decimal to binary and back again, which means that BASICs using binary coded decimal arithmetic are more suitable for business applications. So for business applications, SWTPC BASIC for the 6800 or Whipple and Arnold's BASIC ETC for the 8080 would be more suitable than most other BASIC interpreters.

I own a BASIC SWTPC 6800 and an AC-30 cassette interface, bought at Doc's Computer Shop, 5755 Nolensville Road, Nashville, Tennessee. I have a CT-64 terminal kit and another 4K of memory on order. So far I've gone as far as I can without the terminal (not far at all); all the components I do have are assembled, and I've been able to check them out using Doc's (alias Dave McLennas) ADM-3 terminal. Everything's worked the first time it's been powered up. It must be the equipment, as I'm totally inexperienced in building electronic gear. Or maybe it's that I'm too ignorant *not* to follow instructions. I do know that if it were not for the computer store, I wouldn't be even this far along. Support your local computer store, they can do good things for you. Now if only my terminal would come . . .

Bud Hamblen
Nashville TN 37204

Thank you, Bud. You brought up some interesting points there. The response to Tom and Phil's article has been tremendous. As a matter of fact, they're currently putting together a follow-up article. – John.

Need Article Ideas? Read on . . .

Congratulations! In less

CP-118

CP-119

than half an issue, *Kilobaud* is now the best magazine of its category on the market. Even better than *73*.

Enough with the flowers, let me tell you what a guy like me is expecting from you guys:

1) Technical articles
– description of new projects, parts, pieces of gear, etc.
– basic principles
– I suggest a complete Altair bus description for next issue, and this information should be updated as changes or improvements come along.

2) Projects, construction (hardware). I would like to see construction projects by the car load, like you did in *73* at the beginning, complete with schematics and printed circuit layouts. PC boards should be made available, or at least negatives should be available to amateur computer clubs. I suggest the following projects:
– different CPU cards using the 8080, Z80, 6800, 6502, etc. Altair bus compatible.
– 4K, 8K and 16K byte memory cards, Altair bus compatible, DIP switch to select memory allocation, stand-by power, preferably built around 2102 type chips.
– PROM programmer for UV, EE, or whatever ROMs. A few articles here could give the main piece of equipment, and then once in a while a lots of plug-in modules for different types of ROMs. Software should be available at low cost for popular CPUs.
– Cheapy printer made at home using junk box parts: regular cheap DC motors, steel wire, and a good 5 x 7 impact head should do the job. Same for a plotter.
– Switching power supply for +5V at X amps (many amps).
– MODEM built around Motorola Modem Chips or equivalent, Altair bus compatible.
– Multi-format (Lancaster, KC, Tarbell, CUTS, etc.) audio cassette interface, Altair bus compatible.
Conversion between Altair bus and Digital Group bus, and vice versa would be of great interest: it is a *must* and it is *urgent*.
– Floppy disks & digital transport interfaces, Altair compatible.
– Altair motherboard, at least 8 slots.
– A standard well designed computer front panel (like the one of Cromemco Z1 or Imsai, or Altair 8800B)
– Cabinets
– TV writers
– TV Graphics
– and many more . . .

3) Software
– Listings of different goodies: extended BASIC, FORTRAN, APL, etc. with logical charts, how they are built, how to modify them.
– Keep people informed of new software which is currently being developed and when it is going to be available, and where to write to get information about it.
– Space for software exchange between clubs.
– Algorithms to compute sin, cos, tan, or whatever.
– Small business software.
– More info. on commercial software.
– Explain different things like relocatable assembler, absolute loader, linking loader, etc. and where to put them in memory, so that you get an optimal system.
– Programming tricks to save on memory.
– Comparisons between the instructions of different micros, and how to convert software from one to the other.
– What about real time clock systems?
– Games (I have always dreamed to go to the end of a Monopoly game, and see what will happen, especially while taking a b . . ., yes why not a beer?) tennis on TV, and others.

4) Updated club list

5) Where to find rare parts like acoustical couplers, rubber cushions needed to put in the telephone handset and others.

6) Talk a lot about standards: cassettes, Altair bus and new proposed ones.

CP-120

7) Objective comparisons between different systems and how to interface them (Altair, Digital Group, Ohio, and many others).

8) A service to put clubs on an automatic all info, instead of sending the bingo card each time: it would be a very good ad for manufacturers, and it would mean good business too, as well as one of the best services we can think of from a magazine.

9) A few articles in coming *Kilobaud* issues could talk about:

– Simple things as how to interface a keyboard, any keyboard (SWTP KBD1, 2, or 5, surplus, Digital Group and others) to a micro system.

– Details of DMA, how to do it.

– Definition of handshaking, how to do it.

– Give a list of application notes from manufacturers, of interest.

I will end here for now, and wait for what is coming on in *Kilobaud*. Carry on folks, you are doing a good job. All my friends in our club say and think the same as I do by the way, and there are 40 of us.

Jean-Luc Fontaine
Cap Rouge, P.Q.
Canada GOA 1KO

Thank you, Jean, you've got some fantastic ideas there . . . and you've saved Wayne and I the trouble of having to write them in an editorial . . . again!. – John.

Some Observations On The Industry

I'm in a profession not directly connected with computers or electronics. About nine months ago I stumbled upon, and became fascinated with hobby computers. As a newcomer to the scene, I perhaps have a different perspective than the old-timers (2 years?). I'd like to share a few of my experiences, some observations, and some conclusions:

Because there were then no stores, and there was very little activity in my local area, I began by sending away for as much product literature from manufacturers as I could, and subscribed to four publications. I rapidly learned that one could expect a wide variation in the material received, from manufacturers and publishers alike. The common thread is that in both cases, some of the entrepreneurs are far better technicians than businessmen!

In my product literature search, I got unbelievable responses. In several cases I got *no* response from my requests for firms which had made new product announcements through full page paid advertisements in the publications. Why a firm would bother to advertise, then not respond to a sales lead, totally escapes me. In other cases, the quality of the product brochure or flyer was so poor as to convey the image that the firm was a fly-by-night concern. In one case, a firm sent me a questionnaire asking me what information I wanted (which product lines?), when I had checked specific items on the coupon clipped from their advertisement on my initial inquiry. On the other side of the coin, many firms sent me comprehensive literature, in some cases including product lines far beyond the scope of what I'd asked for. Several sent me, without charge, Catalogs or Microprocessor Manuals which are advertised at prices from $1. to $5. The one suggestion I would offer *all* vendors: Many people blunder into this "hobby computer world" totally ignorant; a twenty-page glossy brochure doesn't really help if the recipient doesn't understand the terminology yet. (The first piece of literature I received was from MITS; I was very impressed, but frankly, at that stage, had no idea what they were talking about!) All suppliers should offer an optional

package with their literature, directed toward the total neophyte, and explaining the fundamentals.

A comment, too, on Computer Stores. A business trip took me to Phoenix in February and I scheduled an extra vacation day prior to the business meeting to give me an opportunity to visit some real live Computer Stores. Imagine my surprise when I discovered that all the Computer Stores in Phoenix are closed on Mondays!

On to publications: Let me say at the outset that *Kilobaud* has proven to be the best, by far, to meet my particular needs. I would especially like to applaud the Kilobaud Klassroom series by George Young which began in issue 5. Just what I need to update my "hardware" knowledge, which has been passed by with the newer technology; equally great for my 13 year old son, who's just beginning.

But let me relate my experience with the other two publications to which I took "Charter Subscriptions": One accepted my subscription order in December, 1976. It had apparently published its first and last issue four months prior to accepting my order. Last month I learned that it had been merged with a second personal computing magazine to which I also hold a Charter Subscription. So I'd paid for two subscriptions and will receive only one publication. Would you believe that the folks at these publications won't even respond to my inquiries regarding a partial refund? It's not being ripped off which infuriates me; it's the incredible lack of business sense of the management of this fledgling publication in ignoring correspondence from a charter subscriber. As a consumer, I've easily concluded which of my subscriptions to continue, and which not to continue.

My whole point is that with manufacturers, retail stores and publications alike, there will be rapid attrition: the fittest will survive, and many will not. And the success/failure criteria will include public relations, marketing skills, and just plain business sense equal to, or even more important than, technical expertise!

Joe Warkany
Cincinnati OH 45208

Some real food for thought, Joe. Thank you. – John.

Byting Comments on Sniping

I want to congratulate you and your staff on the publication of a fine magazine. In my opinion *Kilobaud* is filling a genuine need in the small computer field.

However, it is also my opinion that the continual juvenile sniping at the "other" small systems journal has no place in a serious publication. The June (number 6) issue was the worst in that respect. I also hope that it was the last. The field is big enough for more than one well-edited magazine. Constant references to past differences among publishers, founders, staff, and authors is sophomoric, as are snide references to editorial policies. If indeed the mere existance of a second magazine in Peterborough causes personal pain, keep it personal.

You have to date put together what I would consider a very professional magazine. Not professional in the Time-Newsweek sense, though you do not suffer in any comparison with them, but professional in the programming-engineering sense. I want you to keep that up. I want you to even get better.

Paul J. Gans
New York NY

You know, Paul, you're a real killjoy! Did it ever occur to you that we enjoy picking on that other magazine? – John.

Larry Fish
123 E. Arkansas
Denver CO 80210

Troubleshoot Your Software

... a trace program for the 6502

Here's a rather significant piece of software for you 6502-types. It should make your software debugging a real fun exercise (who am I kidding?). A little note of caution is in order. Larry mentioned that he did assemble the program by hand, therefore the format isn't directly compatible with MOS Technology's Cross Assembler . . . I guess there's only one, right? – John.

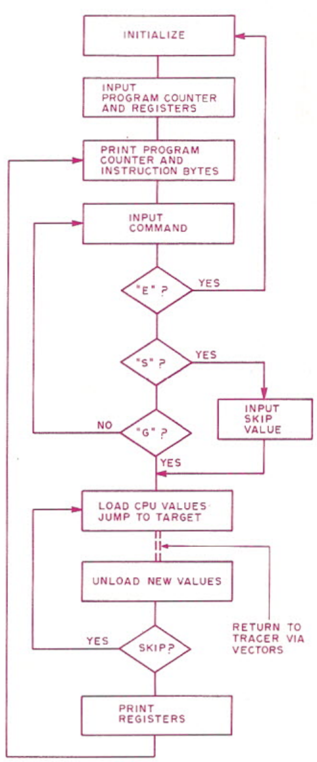


Fig. 1. Program flowchart.

One of the most frustrating parts of programming is the program that doesn't do what it is supposed to do. This is especially true with machine language programs where a single misplaced bit in a forgotten register can destroy a whole program. The usual arsenal of debugging tools include setting break points, examining memory, and careful analysis. These techniques are fine if you have lots of time and patience, but there is a simpler method: Tracer.

Overview

Tracer is a program that provides a simple, straightforward method of debugging faulty programs. It allows you to single-step execute each instruction in a program under software control. Tracer prints each address, each instruction, and after each execution, prints the contents of the registers. All branches and jumps are followed or not followed in exactly the same manner as normal program execution. The Tracer program also allows you to skip ahead, executing any number of instructions at almost full speed. This allows you to skip past subroutines and long program loops.

Most tracer or emulator programs are exceedingly long and complex since they must duplicate in software all hardware functions of the CPU. Large chunks of memory must be used to imitate addressing modes, follow branches and duplicate instructions. Tracer, on the other hand, is simple and requires only 300 bytes of memory.

The secret to Tracer's simplicity is the hardware timer that is available on the TIM, KIM-1, and JOLT ROMs. This timer is tied directly to the CPU clock and can be set to interrupt the CPU anywhere between one and 262,144 clock cycles. Tracer uses this clock to control the CPU's execution of each instruction.

The program uses a dummy Program Counter, Stack Pointer, X, Y and Status Registers. At the beginning of a trace run the user types values into these registers. With each single execution the Program Counter and register values are loaded into the CPU. The timer is then set and the CPU jumps to the target instruction in the program being debugged. At this point, the timer goes off and CPU is interrupted mid-instruction. The processor as a part of its normal interrupt sequency finishes the instruction, saves the program counter and status register, and jumps back to Tracer, which finishes the process of saving the resulting registers. In this way Tracer steps its way through the target program.

Operation

Once Tracer is started it types a P character. This is a request for the starting address and register values of the program to be debugged. The user simply types in these values in the following order: Program Counter, Status Register, Accumulator, X register, Y register, and Stack Pointer. Tracer automatically supplies spaces between each value typed. After the Stack Pointer is typed, Tracer prints the first address and one, two, or three bytes of instruction. Tracer is now ready for a command. The following commands control the execution of the trace sequence: Typing a G command causes the program to execute a single instruction. Typing an E command causes the program to *escape* from the current trace sequence so that new values maybe loaded. Typing an S command allows the program to skip ahead at nearly full speed from the current location to a preset

```
1.  .: 1FC0 00 00 00 00 FF
2.  .G
3.  P 7000 00 00 00 00 FA
4.  7000 85 F9     G 22 00 00 00 FA
5.  7002 A9 23     G 20 23 00 00 FA
6.  7004 D0 55     G 20 23 00 00 FA
7.  705B 85 FE     G 20 23 00 00 FA
8.  705D D8        G 20 23 00 00 FA
9.  705E 4A        G 21 11 00 00 FA
10. 705F 86 FA     E
11. P 7000 00 00 00 00 FA
12. 7000 85 F9     S705B
13.    20 23 00 00 FA
14. 705B 85 FE     G 20 23 00 00 FA
15. 705D D8        G 20 23 00 00 FA
16. 705E 4A        E
17. P↓
18. .
```

Example 1. Tracer execution. User input material is underlined. Line numbers are added for clarification.

location.

Example 1 will clarify the usual trace sequence.

Lines 1 and 2 are the starting procedure used by the TIM and JOLT ROMs. Tracer begins on line 3. When the program is started, Tracer responds with a Carriage Return, a Line Feed, and prints P followed by a space. The user then types in the program counter of the first address in the target program. This is followed by the Status Register, the Accumulator, X register, Y register and the Stack Pointer. As you can see, Tracer is now set to begin execution at address 7000, with all registers set to 00 and the Stack Pointer set to FA. After the Stack Pointer is typed in, the program prints another Carriage Return and Line Feed, then prints the Program Counter of the first location and one, two, or three bytes of instruction. The program is now waiting for a command. Typing a G as illustrated in line 4 causes Tracer to execute a single instruction, print the resulting registers and print the new Program Counter value and instruction bytes. Spacing is automatically set for one, two, and three byte instruction. Lines 4 through 9 illustrate a short trace sequence. Notice that the register values change in response to certain instructions. Also, in lines 6 and 7 Tracer follows an 85 byte forward branch.

Line 11 illustrates the use of an E command to restart the sequence with a new address and/or new register values. When E is typed the program prints P and the user types in the new values.

An S command allows the execution to skip ahead at almost full speed, without printing registers, instruction bytes, or the program counter. This is accomplished (as in line 12) by typing S and then typing the stop location. This *must* be the address of the *first* byte of an instruction, otherwise the program will run away with unpredictable results. If you wish to escape from Tracer back into the TIM or JOLT monitor, type an E command, then a Carriage Return (lines 16, 17 and 18). A period indicates that you are back in the monitor.

Debugging Hints: Naturally, the exact debugging procedure depends on the program and the nature of the fault. Bad programs give clues. They will often execute parts of the routine correctly before blowing up. The usual trace procedure involves starting Tracer at a point in the program that is clearly ahead of the fault. It is then a simple matter to single step until an error is found. If the error is subtle, think out what each instruction is supposed to do; then carefully watch the registers after each instruction.

Branches are a common source of problems. They can be tested by typing an E command and running through the branch with both branch and nonbranch conditions. Check to see that the branch lands on the *first* byte of the intended instruction.

Tedious subroutines involving hundreds of steps away from the main body of the program can be handled using the skip command. By setting Tracer to skip to the first instruction immediately following the subroutine call, Tracer will execute the subroutine at almost full speed and return with all of the registers set by the subroutine.

Input and output routines present special problems for the skip command. For example, if you use the S command to skip through a subroutine that inputs a character from the keyboard, the character loaded will be incorrect. This is caused by the fact that Tracer does not execute at full speed (about 12,000 instructions per second in the skip routine) and time constants used by the input routines are altered. If the correct character must be used to test a portion of a program, it can be hand loaded by resetting a register or loading the character directly into memory. Similar speed-related problems can be handled in the same way.

Tracer is written to run directly on an MOS Technology 6502 TIM or JOLT system having at least 300 bytes of memory starting at location 1F00 hex. It also uses the 55 bytes of unused memory on the 6530 (TIM-JOLT) chip. Adapting the program to KIM-1 systems should be a simple matter since KIM has two timers. Just change the input and output subroutine calls and the timer address. Tracer could be rewritten for any system that has a hardware timer (the flowchart in Fig. 1 should prove an aid in such an effort). For systems that don't have built-in timers, MOS Technology has several timer-port combinations for under $20. Also possible is some kind of TTL flip-flop counter to interrupt the processor after several cycles. Further information on programming and addressing the MOS Technology timer can be found in MOS Technology's hardware and software manuals and in the JOLT manual. ■

Program Listings — Tracer. (Note: Not all zero page locations are consecutive.)

Location	Contents	Label	Inst.	Operand	Comments
00D6	00	XPCL			;Storage for Program Counter
00D7	00	XPCH			;Accumulator, Stack Pointer
00D8	00	XSTA			;and X,Y and Status Reg.
00D9	00	XACC			
00DA	00	XX			
00DB	00	XY			

```
00DC    00              XSP
00DD    01              BITO                        ;Bit Masks
00DE    07              BIT012
00DF    04              BIT2
00E0    08              BIT3
00E1    10              BIT4
00E2    80              BIT7

00E6    FF              FLAG                        ;Flag to indicate multiple
                                                    ;instructions (FF=clear)
00F4    00              STO                         ;Stores #of bytes used by opcode
00F5    00              SPACER                      ;Used to calculate #of spaces
                                                    ;after opcode printout
                                                    ;(Note 00E6, 00F4 & 00F5 are
                                                    ;within TIM's zero page,but
                                                    ;are unused by TIM

                        ;MAIN BODY OF PROGRAM
1F00    A9 9C           BEG     LDA     #9C         ;Initialize vectors and FLAG
1F02    8D F8 FF                STA     UINT
1F05    A9 1F                   LDA     #1F
1F07    8D F9 FF                STA     UINT+1
1F0A    A9 FF                   LDA     #FF
1F0C    85 E6                   STA     FLAG
1F0E    20 8A 72                JSR     CRLF (TIM)  ;Input PC,S,A,X,Y, & SP
1F11    A9 50                   LDA     "P"
1F13    20 C6 72                JSR     WRT (TIM)
1F16    20 77 73                JSR     SPACE (TIM)
1F19    20 A4 73                JSR     RD0A (TIM)  ;Input 2 byte hex address
1F1C    20 DE 1F                JSR     ALPC
1F1F    A0 00                   LDY     #00
1F21    20 77 73        P1      JSR     SPACE (TIM)
1F24    20 B3 73                JSR     RDOB (TIM)  ;Read one hex byte
1F27    99 D8 00                STA(Y)  XSTA        ;Indexed store of reg.
1F2A    C8                      INY                 ;Count
1F2B    C0 05                   CPY     #05         ;Test for 5 reg. loaded
1F2D    D0 F2                   BNE     P1          ;Loop until done
1F2F    20 E5 FF                JSR     PPC         ;Print Program Counter
1F32    B1 D6                   LDA(Y)  XPCL        ;Pick up opcode of 1st
                                                    ;instruction through indirect
                                                    ;pointer
                        ;This routine calculates the number of bytes required
                        ;by each opcode

1F34    F0 25                   BEQ     1BYTE       ;Tests for BRK inst.
1F36    C9 60                   CMP     $60         ;Test for RTS
1F38    F0 21                   BEQ     1BYTE       ;Branch if true
1F3A    EA                      NOP
1F3B    24 E0                   BIT     BIT3
1F3D    F0 0E                   BEQ     HALFOP
1F3F    24 DF                   BIT     BIT2
1F41    D0 16                   BNE     3BYTE       ;Branch if 3 Byte op
1F43    24 DD                   BIT     BITO
1F45    F0 14                   BEQ     1BYTE       ;Branch if 1 Byte op
1F47    24 E1                   BIT     BIT4
1F49    F0 0F                   BEQ     2BYTE       ;Branch if 2 Byte op
1F4B    D0 0C                   BNE     3BYTE       ;Branch if 3 Byte op
1F4D    24 DE           HALFOP  BIT     BIT012
1F4F    D0 09                   BNE     2BYTE       ;Branch if 2 Byte op
1F51    24 E1                   BIT     BIT4
1F53    D0 05                   BNE     2BYTE       ;Branch if 2 Byte op
1F55    24 E2                   BIT     BIT7
1F57    D0 01                   BNE     2BYTE       ;Branch if 2 Byte op
1F59    E8              3BYTE   INX                 ;Count #of Bytes
1F5A    E8              2BYTE   INX
1F5B    E8              1BYTE   INX
1F5C    86 F4                   STX     STO         ;Save count

                        ;This routine prints the opcode and operand
                        ;and a variable number of spaces according to the # of bytes
1F5E    A9 07                   LDA     #07         ;Initial spacer
1F60    85 F5                   STA     SPACER
1F62    B1 D6           P2      LDA(Y)  XPCL        ;Load op index indirect
1F64    20 B1 72                JSR     WROB (TIM)  ;Print it
1F67    C8                      INY                 ;Count
1F68    C4 F4                   CPY     STO         ;Test if done
1F6A    F0 0C                   BEQ     A1          ;Branch when done
1F6C    C6 F5                   DEC     SPACER      ;Subtract three spaces
1F6E    C6 F5                   DEC     SPACER
1F7D    C6 F5                   DEC     SPACER
1F72    20 77 73                JSR     SPACE (TIM)
1F75    4C 62 1F                JMP     P2          ;Loop
1F78    A6 F5           A1      LDX     SPACER      ;Routine prints spaces
1F7A    20 77 73        A2      JSR     SPACE (TIM)
1F7D    CA                      DEX                 ;Count spaces
1F7E    D0 FA                   BNE     A2          ;Loop until done
1F80    20 C0 FF                JSR     COM         ;Input and process commands
1F83    20 77 73                JSR     SPACE (TIM)

                        ;This routine sets up the registers to execute the single
                        ;instruction. The TIM timer is used to interrupt the CPU
                        ;so that only one instruction is executed.
```

```
1F86    A6  DC          P3      LDX      XSP             ;Load stack pointer
1F88    9A                      TXS
1F89    A6  DA                  LDX      XX              ;Load X reg.
1F8B    A4  DB                  LDY      XY              ;Load Y reg.
1F8D    58                      CLI                      ;Prepare for interrupt
1F8E    A9  11                  LDA      #0B             ;Set timer
1F90    8D  0C  6E              STA      TIMER (TIM)     ;Timer will interrupt in 17 cycles
1F93    A5  D8                  LDA      XSTA            ;Load Status
1F95    48                      PHA                      ;Push to save while accum. loaded
1F96    A5  D9                  LDA      XACC            ;Load accum.
1F98    28                      PLP                      ;Load status
1F99    6C  D6  00              JMP      XPCL            ;Indirect to execute one instruct.

                        ;This routine stores the registers after interrupt

1F9C    85  D9                  STA      XACC            ;Store accumulator
1F9E    86  DA                  STX      XX              ;Store X
1FA0    84  DB                  STY      XY              ;Store Y
1FA2    68                      PLA                      ;Store status reg.
1FA3    85  D8                  STA      XSTA
1FA5    68                      PLA                      ;Store PC low
1FA6    85  D6                  STA      XPCL
1FA8    68                      PLA                      ;Store PC high
1FA9    85  D7                  STA      XPCH
1FAB    BA                      TSX                      ;Store Stack
1FAC    86  DC                  STX      XSP
1FAE    A2  FF                  LDX      #FF             ;Set to test skip ahead
1FB0    AD  04  6E              LDA      TIMER (TIM)     ;Clear interrupting timer

                        ;This routine tests to see if we are to skip ahead several
                        ;Instructions

1FB3    E4  E6                  CPX      FLAG            ;Test for skip ahead
1FB5    F0  0F                  BEQ      B2              ;Branch if not
1FB7    A9  00          SET     LDA      #00             ;Program store skip offset here
1FB9    C5  D7                  CMP      XPCH            ;Test offset /c current PC
1FBB    D0  06                  BNE      B1              ;Branch if skip continues
1FBD    A9  00          SETA    LDA      #00             ;Program store offset here
1FBF    C5  D6                  CMP      XPCL            ;Test /c PC
1FC1    F0  03                  BEQ      B2              ;Branch when skip is done
1FC3    4C  86  1F      B1      JMP      P3              ;Execute next
1FC6    86  E6          B2      STX      FLAG            ;Clear Flag
                        ;This routine prints registers

1FC8    A0  00                  LDY      #00             ;Set index
1FCA    B9  D8  00      C1      LDA(Y)   XSTA            ;Load 1st register
1FCD    20  B1  72              JSR      WROB (TIM)      ;Print it
1FD0    20  77  73              JSR      SPACE (TIM)
1FD3    C8                      INY                      ;Count
1FD4    C0  05                  CPY      #05             ;Test for 5 registers printed
1FD6    D0  F2                  BNE      C1              ;Branch until done
1FD8    EA                      NOP
1FD9    EA                      NOP
1FDA    EA                      NOP
1FDB    4C  2F  1F              JMP      ST              ;Do it all again

                        ;This routine stores starting program counter
                        ;from the keyboard
1FDE    A5  EE          ALPC    LDA      TEMPO (TIM)
1FE0    85  D6                  STA      XPCL
1FE2    A5  EF                  LDA      TEMP1 (TIM)
1FE4    85  D7                  STA      XPCH
1FE6    60                      RTS

                        ;Sub-routine to handle commands
FFC0    20  E9  72      COM     JSR      RDT (TIM)       ;Input command
FFC3    C9  47                  CMP      "G"             ;Test for G command
FFC5    F0  1D                  BEQ      T1              ;Branch through if G
FFC7    C9  53                  CMP      "S"             ;Test for S command
FFC9    F0  07                  BEQ      T0              ;Branch if S
FFCB    C9  45                  CMP      "E"             ;Test for E command
FFCD    D0  F1                  BNE      COM             ;Branch if not
FFCF    4C  00  1F              JMP      BEG             ;Re-set everything and restart
FFD2    20  A4  73      T0      JSR      RDOA (TIM)      ;read two hex bytes
FFD5    A5  EF                  LDA      TEMPO+1 (TIM)   ;Places these two in SET & SETA
FFD7    8D  B8  1F              STA      SET+1
FFDA    A5  EE                  LDA      TEMPO (TIM)
FFDC    8D  BE  1F              STA      SETA+1
FFDF    20  8A  72              JSR      CRLF (TIM)
FFE2    86  E6                  STX      FLAG            ;Set Flag
FFE4    60              T1      RTS
FFE5    20  8A  72      PPC     JSR      CRLF            ;Routine prints XPC
FFE8    A5  D7                  LDA      XPCH
FFEA    20  B1  72              JSR      WROB (TIM)
FFED    A5  D6                  LDA      XPCL
FFEF    20  B1  72              JSR      WROB (TIM)
FFF2    20  77  73              JSR      SPACE (TIM)
FFF5    A0  00                  LDY      #00             ;Clears Y for #bytes routine
FFF7    60                      RTS
```

Don Parks
1039 Ivy Lane
Cary NC 27511

Cure that Hot Power Supply

Sooner or later, most microcomputer owners are going to need power supplies — either to power their home brew machine, their development board (KIM-1, etc) or some I/O goody external to the processor.

One handy source for transformers for that power supply is the nearest Radio Shack store. Five of the Radio Shack types have gone into my home brew computers and I/O devices. They're listed in Table 1.

Connections

Figs. 1 and 2 illustrate some of the ways you can connect these transformers.

Precautions

1) Fuse the primary of the transformers.

2) Use 3-wire line cords (so that equipment is grounded).

3) These transformers run warm (*hot*!) if you run them at their rated load. My practice is to load them to 50% to 70% of their ratings (current ratings given with the circuits are 100% ratings). Be sure you have adequate ventilation in your cabinet.

4) A lot of Radio Shack store personnel will hang these transformers by their leads on the pegboard display walls. This practice often results in damage to the transformer; therefore, inspect the transformer *carefully* before you buy it to assure yourself that the leads have not been pulled loose and that the transformer has received no puncture wounds from banging on those pegs.

5) Look carefully at the surface of the tape on the side of the transformer where the 110 V ac leads (black) are. Often, I have found that the connection between the black lead and the transformer winding will have a sharp spur that will poke through the tape insulation — if this is the case, put a layer of tape over the present layer — I have received too many 110 V shocks because of these spurs.

6) Before you hook power to your project, turn on the newly built power supply — verify that the output voltages are correct — then let it run by itself for 3 or 4 hours. My experience has been that bad transformers will burn themselves out in this time even without any loading.

Conclusion

Until I find a better source of *standard* transformers at lower prices I'll continue to use the Radio Shack parts — but I will (DO) observe the precautions already outlined. ■

Cat. No.	Volts	Amps	Size
273-1505	12.6 CT	1.2	2" X 2 3/8" X 1 1/2"
273-1511	12.6 CT	3	2 3/4" X 2 1/4" X 2"
273-1512	25.2 CT	2	2 3/4" X 2 1/4" X 2"
273-1513	12	5	4" X 2" X 2 1/2"
273-1514	18 CT	3.5	4" X 2" X 2 1/2"

Cat. No.	Approximate output (at filter)	Rated Amps	
273-1505	18 V dc no load 16 V dc Loaded	1.2	(Fig. 1.)
273-1511	18 V dc no load 16 V dc Loaded	3	(Fig. 1.)
273-1513	18 V dc no load 16 V dc Loaded	5	(Fig. 1.)

Cat. No.	Approximate output (at filter)	Rated Amps
273-1512	18 V dc no load 16 V dc Loaded	2 (4 in full wave connection) (Fig. 2.)

Note: sum of currents from + and - power supplies must not be greater than 4 amps.

Cat. No.	Approximate output (at filter)	Rated Amps
273-1514	13 V dc no load 11 V dc Loaded	4 (8 in full wave connection) (Fig. 2.)

Note: This transformer is adequate for powering regulators for a +5 supply.

Table 1. Specifications for 5 popular Radio Shack transformers.

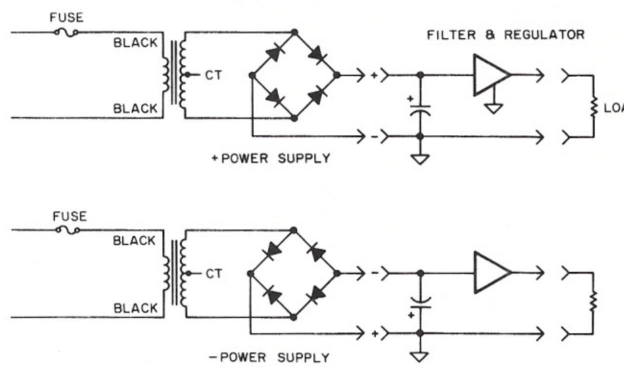


Fig. 1. Full Wave Bridge connections.

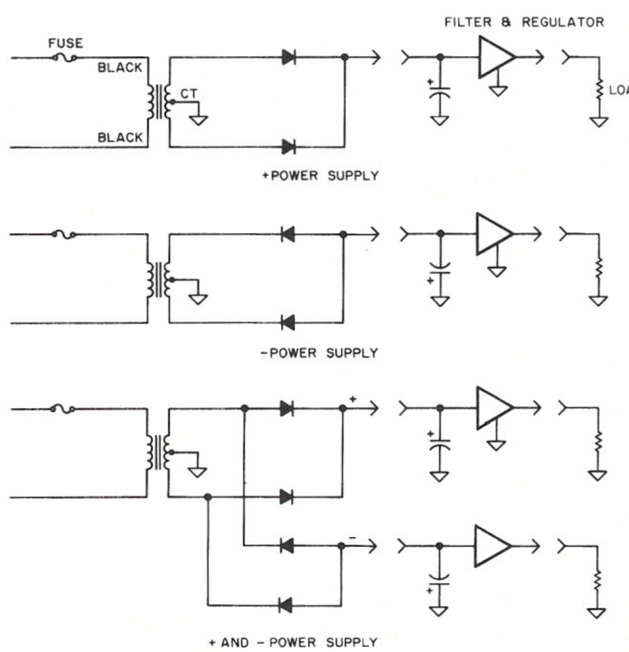
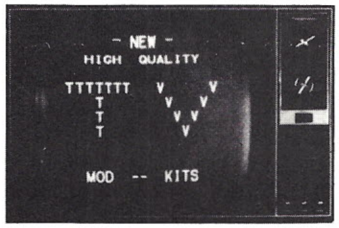


Fig. 2. Full Wave connections.



Authors

Okay, now you see what we're trying to do with Kilobaud — the type of article newcomers to computing want to read. If you've done anything at all with your own system you have the makings of an article which will help someone else through the woods. Write it. Invest a bit of your time at the typewriter and you'll have more money to expand your system. You may even be able to make each addition pay for itself with articles.

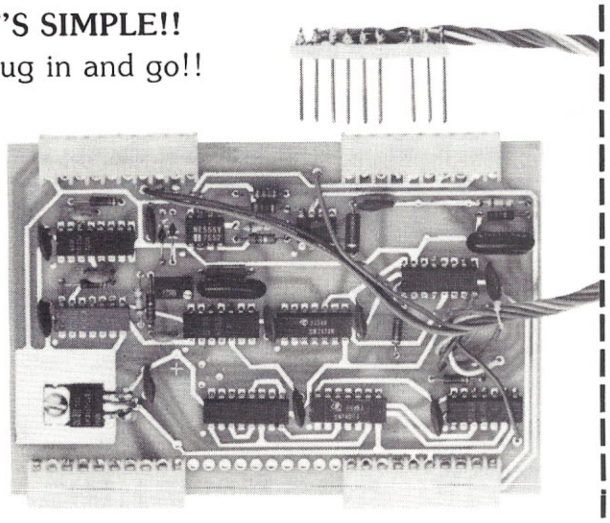
If you need detailed instructions on how to prepare an article drop a note to the Kilobaud Pre-Millionaires Klub, Peterborough NH 03458 and ask.

Readers want to know about everything new on the market . . . how to get things working . . . interfaced . . .any gadgets you've built . . . converted . . . programs you've written. Make their life easier and more fun . . . and get rich and famous as a result. Famous, anyway.

Write, Kilobaud • Peterborough NH 03458

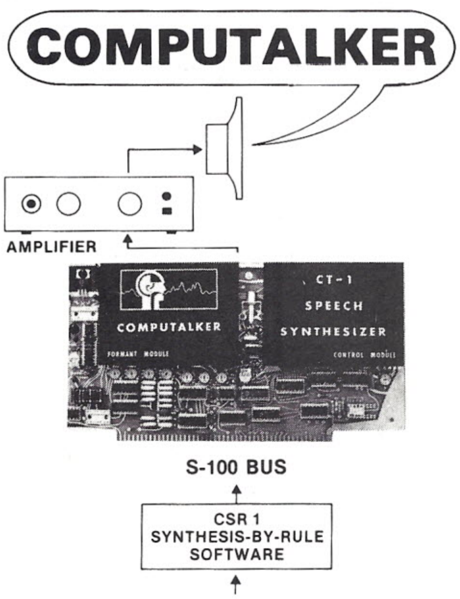
IT'S SIMPLE!!

Plug in and go!!



ACI-33 Cassette Interface

The ACI-33 is a greatly simplified audio cassette interface designed primarily for the Southwest Technical Products Corporation 6800, the control interface and a terminal. The unit will also operate with any RS-232 terminal and computer serial I/O. To use the interface it is only necessary to plug it into an unused I/O slot on the mother board (for power) plug the terminal which was connected to the control interface into the connector provided on the ACI-33 and the connector from the ACI-33 cable into the control interface connector. Priced at $59.95 for an assembled model the interface is available from **Personal Computing Company, 3321 Towerwood Drive Suite 101, Dallas, Texas 75234. Phone [214] 620-2776.** P8



SPEAK "KAAMPYUTAOLKER"

MODEL CT-1 SYNTHESIZER	395.00
CSR1 SOFTWARE SYSTEM	35.00
DEMONSTRATION CASSETTE	2.95

CALIF. RESIDENTS ADD 6% SALES TAX

WRITE FOR INFORMATIVE LITERATURE

COMPUTALKER CONSULTANTS

P.O. BOX 1951, DEPT. K, SANTA MONICA, CA 90406

C36

MICROCOMPUTERS Inc.

539 AMHERST ST. NASHUA NH 03060 (603) 889-1646

HOURS: M,T,W: 10-6 TH,F: 10-9 SAT: 9-6

Almost any computer store can sell you a good, name brand computer, maybe even give you a "great deal." However, when something is missing, broken, or just doesn't work, sometimes these stores are not the same happy, smiling fellas.

Microcomputers Inc. of Nashua NH sells name brand computer products and accessories — IMSAI, Polymorphic, STM, Northstar, Cromemco, more to come. We also sell service and software support to anyone who needs it — not only if you've purchased something from us. We'll also listen to your problems — advice is free. Not all dealers are happy to just shoot the breeze about potential and/or real problems and their possible solutions!

Please stop in and see our store, meet our people, and look over our demo equipment. Play chess with any of our computers. And there's lots more. Stop by sometime — or call — or write a letter — we're here to help you out!

M16

RAINBOW COMPUTING, INC.

Supplier of
Wave Mate
The Digital Group
Digital Equipment Corp.
Computer products

Peripherals and Supplies from

PerSci	**Computer Devices**
Centronix	**Lear-Siegler**
Diablo	**Multi-Tech**
Maxell	**Scotch**

Texas Instruments

Specialists in Design, Implementation and Support of Custom Hardware/Software Systems for Business, Educational, and Personal Use.

Experts in most major computer software including **CDC, IBM, PDP BASIC, COBOL, FORTRAN, PL1 LISP, SIMULA, SNOBOL, SPSS, BMD's COMPASS, MACRO, 6800 & Z80** assembly languages

10723 White Oak Ave.
Granada, Hills CA 91344
(213) 360-2171

R-12

THE COMPUTER CORNER

White Plains Mall, Upper Level
200 Hamilton Ave.
White Plains NY 10601
Phone: (914) WH9-DATA

Near Bronx River Parkway & Cross Westchester Expressway. Plenty of parking.

"The S100 Bus stops at White Plains" with one of the largest collections of boards compatible with the Altair Bus (also IMSAI) in the greater NY area.

You've read about the Sol-20, now come up and see it. We carry Processor Tech, Polymorphic, IMSAI, North Star, TDL, Blast Master and Pickles and Trout.

GOOD PRICE AND SERVICE
10-6 Mon.-Sat.
Thurs. till 9

C-28

THE COMPUTER CORNER

GOOD BOOKS?

If you run across a book which you think other hobbyists would profit knowing about, why not write a brief review . . . including the name and address of the publisher (if you have it) and the price and send it to

kilobaud Peterborough NH

DIODES/ZENERS

1N914	100v	10mA	.05
1N4004	400v	1A	.08
1N4005	600v	1A	.08
1N4007	1000v	1A	.15
1N4148	75v	10mA	.03
1N753A	6.2v	z	.25
1N758A	10v	z	.25
1N759A	12v	z	.25
1N4733	5.1v	z	.25
1N5243	13v	z	.25
1N5244B	14v	z	.25
1N5245B	15v	z	.25

SOCKETS/BRIDGES

8-pin	pcb	.25	ww	.45
14-pin	pcb	.25	ww	.40
16-pin	pcb	.25	ww	.40
18-pin	pcb	.25	ww	.75
22-pin	pcb	.45	ww	1.25
24-pin	pcb	.35	ww	1.25
28-pin	pcb	.35	ww	1.45
40-pin	pcb	.50	ww	1.95

Molex pins	.01	To-3 Sockets	.25
2 Amp Bridge		100-prv	1.20
25 Amp Bridge		200-prv	1.95

TRANSISTORS, LEDS, etc.

2N2222	NPN	.15
2N2907	PNP	.15
2N3740	PNP 1A 60v	.25
2N3906	PNP	.10
2N3054	NPN	.35
2N3055	NPN 15A 60v	.50
T1P125	PNP Darlington	.35
LED Green, Red, Clear		.15
D.L.747	7 seg 5/8" high com-anode	1.95
XAN72	7 seg com-anode	1.50
FND 359	Red 7 seg com-cathode	1.25

C MOS

4000	.15
4001	.20
4002	.20
4004	3.95
4006	1.20
4007	.35
4008	1.20
4009	.30
4010	.45
4011	.20
4012	.20
4013	.40
4014	1.10
4015	.95
4016	.35
4017	1.10
4018	1.10
4019	.70
4020	.85
4021	1.35
4022	.95
4023	.25
4024	.75
4025	.35
4026	1.95
4027	.50
4028	.95
4030	.35
4033	1.95
4034	2.45
4035	1.25
4040	1.35
4041	.69
4042	.95
4043	1.25
4044	.95
4046	1.50
4049	.80
4050	.60
4066	1.35
4069	.40
4071	.35
4082	.45

— T T L —

7400	.15
7401	.15
7402	.20
7403	.20
7404	.15
7405	.25
7406	.35
7407	.55
7408	.25
7409	.15
7410	.10
7411	.25
7412	.30
7413	.45
7414	1.10
7416	.25
7417	.40
7420	.15
7426	.30
7427	.45
7430	.15
7432	.30
7437	.35
7438	.35
7440	.25
7441	1.15
7442	.55
7443	.85
7444	.45
7445	.80
7446	.95
7447	.95
7448	.95
7450	.25
7451	.25
7453	.20
7454	.25
7460	.40
7470	.45
7472	.45

7473	.25
7474	.35
7475	.35
7476	.30
7480	.55
7481	.75
7483	.95
7485	.95
7486	.30
7489	1.35
7490	.55
7491	.95
7492	.95
7493	.40
7494	1.25
7495	.60
7496	.80
74100	1.85
74107	.35
74121	.35
74122	.55
74123	.55
74125	.45
74126	.35
74132	1.35
74141	1.00
74150	1.00
74151	.75
74153	.95
74154	1.05
74156	1.15
74157	.65
74161	.85
74163	.95
74164	.60
74165	1.50
74166	1.35
74175	.80

74176	1.25
74180	.85
74181	2.75
74182	.95
74190	1.75
74191	1.35
74192	1.65
74193	.85
74194	1.25
74195	.95
74196	1.25
74197	1.25
74198	2.35
74221	1.00
74367	.85
75108A	.35
75110	.35
75491	.50
75492	.50
74H00	.25
74H01	.25
74H04	.25
74H05	.25
74H08	.35
74H10	.35
74H11	.25
74H15	.30
74H20	.30
74H21	.25
74H22	.40
74H30	.25
74H40	.25
74H50	.25
74H51	.25
74H52	.15
74H53J	.25
74H55	.25

74H72	.55
74H101	.75
74H103	.75
74H106	.95
74L00	.35
74L02	.35
74L03	.30
74L04	.35
74L10	.35
74L20	.35
74L30	.45
74L47	1.95
74L51	.45
74L55	.65
74L72	.45
74L73	.40
74L74	.45
74L75	.55
74L93	.55
74L123	.55
74S00	.55
74S02	.55
74S03	.40
74S04	.35
74S05	.35
74S08	.35
74S10	.35
74S11	.35
74S20	.35
74S40	.25
74S50	.25
74S51	.45
74S64	.25
74S74	.40
74S112	.90
74S114	1.30

74S133	.45
74S140	.75
74S151	.35
74S153	.35
74S157	.80
74S158	.35
74S194	1.05
74S257(8123)	.25
74LS00	.45
74LS01	.45
74LS02	.45
74LS04	.45
74LS05	.55
74LS08	.45
74LS09	.45
74LS10	.45
74LS11	.45
74LS20	.40
74LS21	.25
74LS22	.25
74LS32	.40
74LS37	.40
74LS40	.55
74LS42	1.75
74LS51	.65
74LS74	.75
74LS86	.75
74LS90	1.30
74LS93	1.00
74LS107	.95
74LS123	1.00
74LS151	.75
74LS153	1.20
74LS157	.85
74LS164	1.90
74LS367	.85
74LS368	.70

9000 SERIES

9301	.85
9309	.35
9322	.85
95H03	.55
9601	.75
9602	.50

MEMORY CLOCKS

74S188 (8223)	3.00
1702A	7.95
MM5314	3.00
MM5316	3.50
2102-1	1.75
2102L-1	1.95
TMS6011NC	6.95
8080AD	15.00
8T13	1.50
8T23	1.50
8T24	2.00
2107B-4	4.95

LINEARS, REGULATORS, etc.

8266	.35
8836	.95
MCT2	.95
8038	3.95
LM201	.75
LM301	.25
LM308 (Mini)	.75
LM309H	.65
LM309K(340K-5)	.85
LM310	1.15
LM311D(Mini)	.75
LM318 (Mini)	.65

LM320K5 (7905)	1.65
LM320K12	1.65
LM320T12	1.25
LM320T15	1.65
LM339	.95
7805 (340T-5)	.95
LM340T-12	1.00
LM340T-15	1.00
LM340T-18	1.00

LM340T-24	.95
LM340K-12	2.15
LM340K-15	1.25
LM340K-18	1.25
LM340K-24	.95
LM373	2.95
LM380	.95
LM709(8,14 PIN)	.25
LM711	.45

LM723	.50
LM725	1.75
LM739	1.50
LM741 8-14	.20
LM747	1.10
LM1307	1.25
LM1458	.95
LM3900	.50
LM75451	.65
NE555	.50
NE556	.95
NE565	.95
NE566	1.75
NE567	1.35
SN72720	1.35
SN72820	1.35

INTEGRATED CIRCUITS UNLIMITED

7889 Clairemont Mesa Blvd. San Diego, CA 92111 (714) 278-4394

All orders shipped prepaid — No minimum

Open accounts invited — COD orders accepted

Discounts available at OEM Quantities

California Residents add 6% Sales Tax

24 Hour Phone (714) 278-4394 — **MasterCharge / BankAmericard**

MICROCOMPUTER

PROM'S

1702A	256 x 8 Bit 1 us TS Erasable	5.00
1702AL	256 x 8 Bit 1 us TS Eras Lo Pwr	7.00
2704	512 x 8 Bit 450 ns TS Erasable	20.00
2708	1024 x 8 Bit 450 ns TS Erasable	27.08
3601	256 x 4 OC 70 ns	4.50
5203AQ	256 x 8 Bit 1 us TS Erasable	7.00
5204AQ	512 x 8 Bit 1 us TS Erasable	10.00
82S23B	32 x 8 Bit 50 ns OC	4.00
82S129B	256 x 4 Bit 50 ns TS	4.25
8223B	32 x 8 Bit 50 ns OC	4.00

WAVEFORM GENERATOR

8038	VCO	4.50
MC4024	Dual VCO	2.75
566	VCO-Function	2.00

CHARACTER GENERATORS

2513	5x7 5 line Lower Case	6.75
2513	5x7 5 line Upper Case	6.75
MCM6571	128 x 9 x 7 ASCII Shifted Greek	10.80
MCM6571A	128 x 9 x 7 Downcount	10.80
MC6572	128 x 9 x 7 ASCII Non-Shifted W/G	10.80

SHIFT REGISTERS DYNAMIC

1404AN	1024 x 1 Bit 2.5 MHz	3.00
2505K	512 x 1 Bit 2.5 MHz	3.00

STATIC

MM506	100 x 2 Bit	.89
2509K	50 x 2 Bit 1.5 MHz	1.00
2518B	32 x 6 Bit 2.0 MHz	3.95
2533V	1024 x 1 Bit 1.5 MHz	2.00
TMS3002	50 x 2 Bit 1.0 MHz	1.00
TMS3112	32 x 6 Bit 2.0 MHz	3.95
MM5058	1024 x 6 Bit 1.5 MHz (8 pin)	2.00

MISC OTHER COMPONENTS

NH0025CN	Dual Low Cost MOS Clock Driver	1.75
NH0026CN	5 MHz Dual Mos Clock Driver	3.00
N8T20	Bi-Directional One Shot	4.00
N8T26	Quad Bus Driver/Receiver	3.25
N8T97	Tri State Hex Buffer	1.45
DM8098	Tri State Hex Inverter	1.00
1488	RS232 Quad Line Driver	1.95
3205	1-of-8 Decoder 18 ns Delay	6.20
D-3207A	Quad NAND to MOS Driver	2.50
C-3404	6 Bit Latch 12 ns O/P Delary	3.95
P-3408A	Tri State Hex MOS Sense Amp	6.75
P-4201	Clock Generator	4.95
MM-5320	T V Camera Sync Generator	6.00
MM-5369	Oscillator Pre-Scaler	2.00
DM-8130	Ten Bit Comparator	2.25
DM-8131	6 Bit Comparator	2.35
DM-8831	4 Input AND NAND Tri State	2.50
DM-8833	Quad Tri State Transceiver (True)	2.50
DM-8835	Quad Tri State Transceiver (Inv)	2.50

U A R T S

AY5-1013	(GI)	6.95
TR-1602A	(WD)	6.95
TMS-6011	(TI)	6.95

SUPPORT DEVICES

3851	Program Storage Unit (F-8)	14.95
3853	Static Memory Interface (f-8)	14.95
3881	Parallel I/O Controller (Z-80)	15.95
3882	Counter Timer Circuit (Z-80)	15.95
TMS5501	I/O Controller	24.99
8212	8-Bit I/O Port	4.25
8214	Priority Interrupt Control	12.95
8216	Bi-Directional Bus Driver	5.25
8224	Clock, Generator & Driver	6.00
8228	System Controller & Bus Driver	9.25
8238	System Controller & Bus Driver	8.20
8251	Programmable Communication Int	12.00
8255	Programmable Peripheral Interface	12.00
8257	Progr Direct Mem Access Control	12.00
8259	Programmable Interrupt Controller	22.00

C.P.U. 'S

8008-1	8 Bit CPU	8.75
8080A	Super 8008	16.95
Z80	CPU (3880)	39.95
F8	CPU (3850)	19.95
CDP1802CD	CPU (RCA)	29.50

DYNAMIC RAMS

1103	1024 x 1 Bit 300 ns	1.50
2107B	4096 x 1 Bit 200 ns	4.50
2107B-4	4096 x 1 Bit 270 ns	4.00
4050NL	4096 x 1 Bit 300 ns	4.50
4096	4096 x 1 Bit 300 ns	5.00
MM5262	2048 x 1 Bit 365 ns	3.00
MM5270	4096 x 1 Bit 200 ns (18 Pin)	5.00
5280	4096 x 1 Bit 200 ns (16 Pin)	6.00

STATIC RAMS

21L02-1	1024 x 1 Bit 350 ns TS	1.58
31L01	16 x 4 Bit 110 ns OC	2.00
1101A	256 x 1 Bit 1 us TS	1.00
2101	256 x 4 Bit 1 us TS	3.00
2102	1024 x 1 Bit 1 us TS	1.25
2102-1	1024 x 1 Bit 500 ns TS	1.50
2111A-4	256 x 4 Bit 450 ns TS (18 pin)	4.45
2112A-4	256 x 4 Bit 450 ns TS (16 pin)	3.00
2501B	256 x 1 Bit 1 us	1.45
3107	256 x 1 Bit 80 ns OC	2.95
4200A	4096 x 1 Bit 250 ns	13.75
74C89	16 x 4 Bit 280 ns TS	3.00
74S201	256 x 1 Bit 50 ns TS	4.75
91L02A	1024 x 1 Bit 500 ns TS	2.00
7489	16 x 4 Bit 60 ns OC	2.25
8225	16 x 4 Bit 50 ns OC	1.50
8599	15 x 4 Bit 50 ns TS	1.50
82S09	64 x 9 Bit	11.00

FIFO

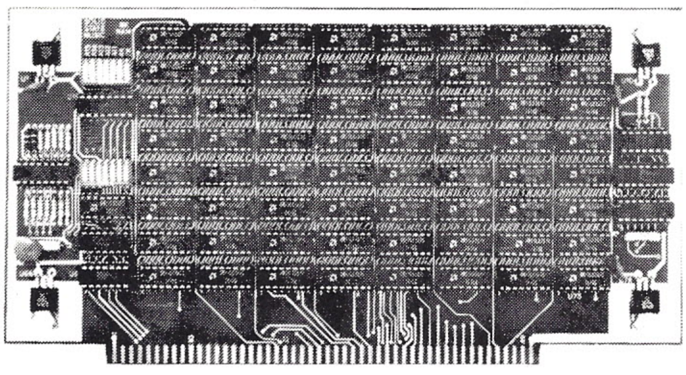
3341A	64 x 4 Bit 1.0 MHz	6.75

JADE CO. OFFERS SPECIAL OF THE MONTH

2708 E-PROM Price $27.08 ea.

8K RAM BOARD KIT
$189.95

©1977



FEATURES

- Plug compatible with the ALTAIR 8800 and IMSAI 8080, or any other system using the "ALTAIR bus."
- Low-power, 350 nanosecond RAMs. No wait cycles required.
- Low-power Schottky support chips.
- DIP switch selection of memory address assignment and wait cycles.
- Memory protect can be set for increments of 256 bits, 512 bits, 1K, 2K, 4K or 8K by DIP switch.
- T.I. low profile sockets provided for all RAMs and ICs.
- Gold plated edge connector contacts.

master charge THE INTERBANK CARD

your BANKAMERICARD welcome

CONNECTORS

EDGE CONNECTORS

10	Dual .156 W W Gold	50
15	Dual .156 W W Gold	75
22	Dual .156 Solder Gold	2.25
50	Dual .125 Solder Gold	5.00
50	Dual .125 Solder Gold (For IMSAI)	3.50
50	Dual .156 Solder Gold (For Altair)	5.00

SUBMINITURE CONNECTORS for RS232

25	Solder type DB(25 S) (Female)	3.90
25	Solder type DB(25 P) (Male)	3.50
	Male & female (25 P & 25 S)	6.50
	Cover Each	1.50

JADE CO

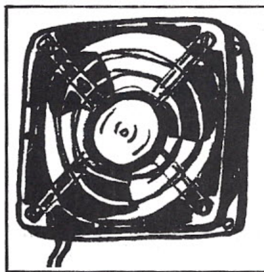
Electronics for the Hobbiest and Experimenter

5351 WEST 144th STREET
LAWNDALE, CALIFORNIA 90260
(213) 679-3313

SOCKETS

I C SOCKETS

8	Lo Pro Tin	.20
10	Transistor IC 10 Gold	.25
14	Lo Pro Tin	.25
14	Wire Wrap Gold	30
16	Lo Pro Tin	.25
16	Standard P C Tin	.30
16	Wire Wrap Tin	.30
18	Lo Pro Tin	.25
18	Wire Wrap Tin	.85
20	Wire Wrap Tin	.95
22	Lo Pro Open Frame Tin	.50
22	Wire Wrap Tin	.95
24	Standard P C Tin	.50
24	Wire Wrap Tin	.85
24	Lo Pro (Open Frame) Tin	.40
28	Lo Pro (Open Frame) Tin	.40
28	Standard P C Tin	.60
28	Wire Wrap Tin	95
40	Lo Pro (Open Frame) Tin	.50
40	Wire Wrap Tin	1.65



rotron whisper fan

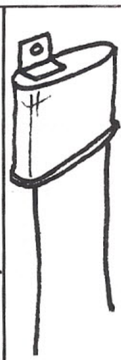
ROTRON WHISPER FANS, 115 AC, 7 Watt fans. 60 CFM. These fans have been removed from equipment, and are fully guaranteed. A chance to keep your equipment cool at low prices. 4½"x4½"x1½".

STOCK NO.5520K $6.95 ea, 2/12.00

TV videocube module

VIDEOCUBE is a fully self contained RF oscillator and modulator, used for interfacing TV games, video cameras, and the video output of mini & micro computers to your TV set. It is compatible with TTL, DTL and CMOS logic. VIDEOCUBE meets FCC requirements. We supply all data for kit assembly, plus reprint of article in Aug.'77 in RADIO-ELECTRONICS by Glen Dash, on The VERSATILE VIDEOCUBE.

STOCK NO.5500K, Complete kit with data $13.95 2/26.00
STOCK NO.5500PK partial kit, all hard to find parts $11.95 2/22.00



crystals

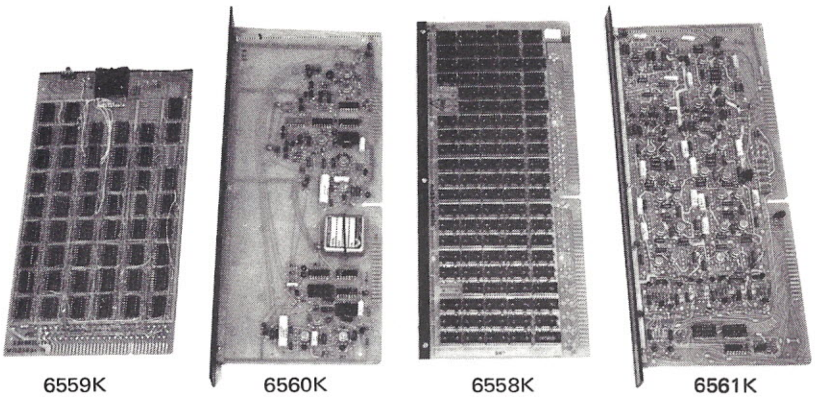
We have a nice assortment of precision crystals, useful for computer timing boards, digital clocks, digital automobile clocks, frequency standards etc.

FREQ.	CASE	LEADS	STOCK NO.
116KHz	HC-6/U	Wire	1013K
158.4KHz	HC-6/U	Wire	1016K
500 KHz	HC-6/U	Wire	1018K
667KHz	HC-6/U	Wire	1014K
3.5 MHz	VM-6	Wire	1015K
3.8 MHz	VM-6	Wire	1012K
4.0 MHz	HC-6/U	Wire	1011K
5.0 MHz	HC-6/U	Wire	1008K

The frequency of these crystals have all been chosen so that they divide down to 50 Hz, 60 Hz or 1 Hz. This makes them ideal for all types of timing and clock devices. We supply data, showing the proper SN74--- ICs to use to get these frequencies.

all types $3.95ea. 2/$7.00

prototype boards op-amp boards crystal timer boards



6559K 6560K 6558K 6561K

We have assembled a fantastic lot of boards that have tremendous appeal for computer buffs, and any builder of electronic equipment. These boards are all out of an operating computer system that has been upgraded.

Board 6558K is a prototype board, that has from 75 to 100 wire wrap IC sockets, both 14 pin and 16 pin. 40 of the 14 pin sockets are in 16 pin holes, so that they may be replaced with 16 pin sockets if you need them. The sockets are at present wire wrapped, so you must remove the wiring by unwrapping it, but when you do you have a board worth over $175.00. Board is 13½"x6", has ground plane on one side, and Vcc plane on other. Ther are 128 gold plated edge contacts.

6559K is another wire wrap proto type board. It has a minimum of 50 sockets, 8 of which are 16 pin, the rest 14 pin. If more 16 pin are needed, the board is drilled for 16 pin sockets, so that the 14 pin sockets may be removed and replaced with 16 pin DIP sockets. The board is unique in that the wire wrap terminals are brought out to the top of the board, rather than the reverse side as the 6558M above. This board also is wire wrapped, and the preveious wiring must be unwound. The board contains a 6 position thumb wheel switch, and a SPST slide switch. There are 70 gold plated edge contacts, and board has a ground plane and a Vcc plane. 11½"x6"

7560K is a clock timing board. It contains a VECTRON CO-231T crystal oscillator including tuning option for an accuracy of .0001% Crystal frequency is 4.9152 which divides conveniently to 60 Hz with 3 SN7493, a SN7490, and a SN7470. It divides to 50 Hz. with 3 SN7493, 1 SN7492, and a SN7470. It goes to 1 Hz. with 3 SN7493, 1 SN7492 and 2 SN7490. This is a very flexible oscillator which can be used for any digital clock including clocks for automobiles, computer clocks etc. The crystal oscillator is plugged into a board that was used for timing, and contains many op-amps, transistors, SN7400 series ICs, dipped tantalum capacitors,plus many other components, more than enough to make your own timing board.

The latest catalog we have from VECTRON is dated 1972, and shows this oscillator at $75.00, plus $10.00 for the tuning option.

6561K is a parts board which contains 31 LM741 OP-AMPS., 32 transistors, over 75 1% precision resistors, a dozen or more dipped tantalum capacitors of various values, plus resistors, capcitors and diodes.

STOCK NO.6558K	75 to 100 socket proto type board	18.75	2/35.00
STOCK NO.6559K	50 Socket proto type board	11.75	2/22.00
STOCK NO.6560K	Crystal Oscillator board	16.95	2/32.00
STOCK NO.6561	Op—Amp Board	10.95	2/20.00

5 Volt 6 amp, ±12 Volt 2 Amp HIGHLY REGULATED POWER SUPPLY

Once in a while we get hold of a supply that is ideal for computers. This one is just that. It has 3 voltages, 5 volts @ 6 Amps, and plus and minus 12 volts @ 2 Amps. Thus you have the necessary voltages for TTL chips, (5 volts) , op—amp, transistor and CMOS (12 volts.) The supply is 11"x5"x4½", and weighs 12 lbs. is fully fused, has a 5½' 3 wire power cord, is fully tested and guaranteed. These supplies were removed from CODEX MODEMS, where no cost was spared to make the best possible supply.

STOCK NO.5519K CODEX POWER SUPPLY, $27.50 ea. 2/50.00

VISIT our 2 new retail outlets: DELTA ELECTRONIC HOBBIES, 5151 Buford Hwy. Doraville, (Atlanta) Ga. and DELTA ELECTRONICS, 590 Commonwealth Ave. Boston, Mass.

DELTA ELECTRONICS

P.O. BOX 2,AMESBURY MA 01913
Phone (617) 388—4705

MINIMUM ORDER $5.00. Include sufficient postage, excess refunded. Send for new Catalog 18, bigger than ever.

BANKAMERICARD and MASTERCHARGE now accepted, minimum charge $15.00. Please include all numbers. Phone orders accepted.

D13

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

TERMS: Add 50¢ orders under $10. Allow up to 5% for shipping; excess refunded. We require street address for COD. BankAmericard® /Mastercharge® ($15 min) call **415-562-0636**, 24h. CA res add tax.

NEW 4K EconoROM $265

with editor, assembler, and monitor routines for the 8080

Improved listing, bigger and better---SP and PSW are now implemented for greater user facility. Very low power, quality board, sockets for all ICs, and so on. Find out what moved Jay Bell, editor of Print-Out, to write:

"Without a doubt, this board is the best buy I have made for my Altair 8800 out of the $5000+ I've spent." PRINT-OUT, September 1976

CPU Power Supply $45

Gives 5V @ 4A with crowbar overvoltage protection, +12V @ ½A, -12V @ ½A, and an adjustable 5-10V bias supply. Although intended for small computer systems, this is also a dandy little bench supply for digital experiments.

COMPUTER KITS

8K EconoRAM II™ $169.00

Configured as two independent 4K blocks for maximum flexibility. Full buffering; tri-state outputs drive the S-100 bus or any bidirectional bus; low power Schottky support ICs plus selected RAMs give low power operation; zips along at 450 ns or better (use 1 wait state with Z-80). Low profile sockets included for all ICs. Plate through, double-sided epoxy glass board.

10 Slot Motherboard $89

Includes 10 edge connectors; use as IMSAI add-on or for stand alone system. Active, regulated terminations minimize crosstalk, overshoot, and other bus problems. Epoxy glass board, quality parts, S-100 compatible, heavy power traces.

18 Slot Motherboard $124

Same features as above, but with 18 slots + edge connectors.

Terminator Board $29.50

Add active terminations to your bus, and clean up the noise, crosstalk, ringing, and overshoot that can foul up data and crash programs. Plug into 1 slot of an S-100 compatible motherboard, and you are ready to go.

DEALER INQUIRIES INVITED

G4

K-rations for your computer

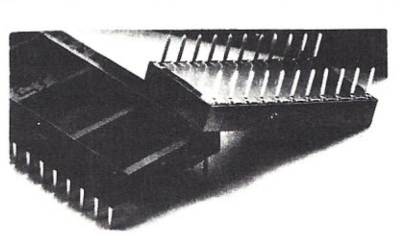
Place BAC/MC orders toll-free to 800-648-5311.
Or see it at your local computer shop.

K-Ration™ 8Kx8 memory with SynchroFresh™.
$188 assembled and warranted.

Now you can load your Altair, IMSAI, Equinox 100 or other S-100 buss computer with 8K x 8 memory boards for just $188 apiece . . . and that's assembled, tested and warranted for repair or replacement for 1 full year. It's possible because Morrow's Micro-Stuff has developed **Synchro-Fresh™**, the first and only memory refreshing system that weaves itself invisibly into the natural timing of the S-100 buss. And that makes the **K-Ration™ 8Kx8 memory** refreshingly reliable and helps keep the cost down. Just $188 assembled with 1 year warranty, kit just $159 (Cal. res. add tax). Postpaid from ThinkerToys™.

K-Ration™ 4Kx8 MEMORY is now the lowest-cost 4K memory available for S-100 buss personal computers. A complete memory board kit, just $109 (Cal. Res. add tax). Postpaid from ThinkerToys™. Product of Morrow's Micro-Stuff

Thinker Toys™ 1201 10th St.
Berkeley, CA 94710



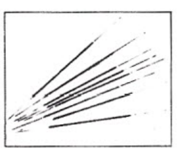
High quality sockets for IC's and PC interconnections. Check our price and quality and you will see why TRI-TEK is fast becoming the leader in IC sockets.

Low Profile DIP Solder Tail (Tin)

		1-9	10-24	25-100
SKT-0802	8 pin	.15	.15	.14
1402	14pin	.18	.17	.16
1602	16pin	.20	.19	.18
1802	18pin	.27	.26	.25
2002	20pin	.29	.28	.27
2202	22pin	.35	.34	.33
2402	24pin	.36	.35	.34
2802	28pin	.42	.41	.40
4002	40pin	.60	.57	.53

3 Level Wire Wrap Gold

	1-9	10-24	25-100
SKT-1400	.38	.37	.36
1600	.42	.41	.40
1800	.73	.65	.59
2400	1.00	.91	.83
4000	1.69	1.51	1.37



PRESTRIPPED WIRE WRAP WIRE

Highest quality 30 ga. Kynar insulated silver plated wire for wrapping. Stripped 1" on both ends. Indicated lengths are lengths of insulated portion. Packed 100 per sturdy plastic vial or 1000 per poly bag. Compare our prices!!!. Available in Black, Red, Yellow and Green. State color desired.

Length	Price per tube of 100	Price per bag of 1000
1"	$1.48 (WW30VC-1)	$11.84 (#WW30BK-1)
2"	$1.60 (WW30VC-2)	$12.80 (#WW30BK-2)
4"	$1.85 (WW30VC-4)	$14.80 (#WW30BK-4)
6"	$2.20 (WW30VC-6)	$17.60 (#WW30BK-6)

ROLLS OF WIRE SAME AS ABOVE (30 ga. KYNAR)
100 ft...$2.95 500ft....$8.95 1000ft...$14.95

WRAP WIRE SPECIAL FOR AUGUST

Special purchase of quality KYNAR insulated 30 ga. wire brings you a real bargain in pre-stripped wrapping wires. Available in blue color only. 1" and 2" insulation only.
1" insulation, blue, bag of 100 pieces..............$.99
2" insulation, blue, bag of 100 pieces..............$1.19

RIBBON CABLE IC INTERCONNECTS

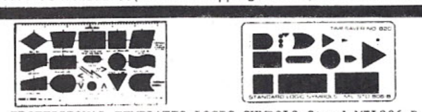
SINGLE END

No Of Pins	6"	12"	18"	24"	36"	48"
14P	1.51	1.62	1.72	1.83	2.05	2.26
16P	1.64	1.76	1.87	1.99	2.21	2.44
24P	2.49	2.69	2.88	3.08	3.48	3.87
DOUBLE END						
14P	2.76	2.87	2.97	3.08	3.30	3.51
16P	3.01	3.13	3.24	3.36	3.58	3.81
24P	4.55	4.75	4.94	5.14	5.54	5.93

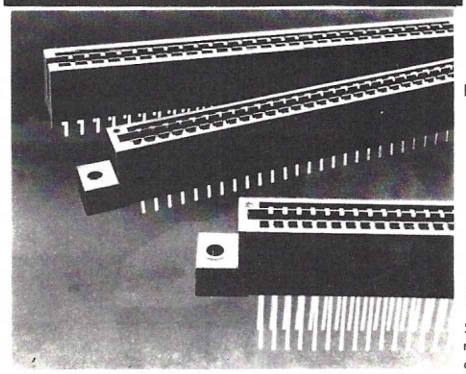
OK WIRE WRAP TOOLS
BW-630 GUN.....$34.95
HW-30 Tool......$5.95
HW-30M Tool.....$6.95
FREE, 50 roll of wire wrap wire with purchase of tool!

NEW BOOK

1977 IC MASTER. Latest edition of this classic reference work has 1263 pages of technical data, cross-references, second source listings, index of available application notes from the guys who make the parts! A free up-date sevice card to help keep your copy current is included. Beautifully bound in hard cover leather grained jacket. Want IC info ? Here 'tis.....
1977 IC MASTER..(includes shipping in USA)........$38.88



PROFESSIONAL TEMPLATES-LOGIC SYMBOLS-Stand MIL806-B
3/4 Size......................................$3.50
1/2 Size.....(Handy shirt pocket size)......$3.25
Computer Flow Chart Symbols................$3.50



100 PIN MINICOMPUTER PC CONNECTORS

2X50 with .125" spacing. Solder tail or wrap terminals. By TI.
PCC-100ST (solder)....$4.99 PCC-100WW....$4.99 4/$17.75

SIGNETICS 8000 SERIES TTL LOGIC

These quality units are faster and have greater fan-out capability than standard TTL. From a giant factory change-over you get real bargain prices. All are house numbered, but we provide a reference and pin-out sheet.
N8880A...Quad-2 input NAND gate...........8/$1.00
N8822A...Dual J-K master/slave F/F........4/$1.00
N8885A...Quad 2-input NOR gate............8/$1.00
N8890A...Hex Inverter.....................6/$1.00
N8202A...10 bit "D" type register.......... $1.25

The following items are available in large quantities Dealer or manufacturer inquiry is invited.
2N3414.. NPN switch on reels...............10/$1
1N753A.. 6.2V, 5%, ¼W Zener................ 5/$1
C106F2.. 50V, 4A SCR w/socket.............. 3/$1
1N967B.. 18V, 5%, ¼W Zener................. 5/$1
42501-1 Quad Hi speed NPN transistor in 14 pin DIP package. Similar to Motorola MPQ3303....... 5/$1

MCM6571A is an 8192-Bit Horizontal-Scan (Row select) character generator with shifted characters. It contains 128 characters in a 7X9 matrix, and has the capability of shifting certain characters that normally extend below the baseline, such as j,y,g,p and q. A 7-bit address code is used to select one of the characters.

Features:
- .Static operation
- .TTL compatability
- .CMOS compatability (5V)
- .Shifted character compatability
- .Includes Greek alphabet
- .Maximum access time =500nS

(See article in March '77 issue of 73 Magazine for applications including TV-Computer interface)
MCM6571A.................................. $9.95
Specs..................................... $1.00

MM5320 TV SYNC GENERATOR I.C.

Generate all the sync pulses necessary for camera or video terminals. Use with MCM6571A in the TV-Computer interface. MM5320N............................$18.80
Specs..60¢

7 SEGMENT TO BCD DECODER, OR IS IT ENCODER?

Think about it-- how many times have you seen an application for device with 7 segment readout if only you had the output in BCD? Calculators, clocks, timers and the like can now be read into your mini'puter with minimum conversion hassel. CMOS for low power drain. Latched.
MM74C74C915N...18 pin DIP.................$2.99

LM1815 ADAPTIVE SENSE AMP CHIP.

Used with motor control to adapt to variable input and noise levels. Applications include zero crossing switch, motor control, tachometers, motor testing.
LM1815N....................................$5.72
Specs.. 30¢

1N5393 200V, 1.5A Diode. Sturdy replacement for 1N4003 at a good savings....................15/$1

2N2369 High Speed NPN switching transistor. Marked for a giant computer company with house number. TO-18 metal case. PRIME PARTS!!!!
#2501-12....(2N2369 House Numbered).......... 8/$1

LM7812KC/LM340-12. Brand new, industrial excess inventory brings you a real bargain in this +12V, 1A regulator. Made by Silicon General for one of those big 'puter companies.
LM7812KC/340-12.......................... $1.49!!

600V, 3 A BRIDGE RECTIFIER

½" hexagonal metal jacket stud mount with 2" stab-on wire terminals. #KBS06..(Mfd. by G.I.)................89¢

MC14412VP Universal Modem Chip $16.95

"COOK BOOKS"

The famous Howard Sams "Cook Book" series tells you what and how in a broad range of subjects. Probably the most widely referenced works in their fields.
Prices quoted include shipping (U.S. only)
TTL Cookbook.............328 pages........$9.95
Active Filter Cookbook......223 pages....... $15.95
CMOS Cookbook...........402 pages........ $10.95
IC OP AMP Cookbook.......579 pages........ $14.25
TV Typewriter Cookbook.....250 pages........ $10.95

MC14411 BIT RATE GENERATOR.

Single chip for generating selectable frequencies for equipment in data communications such as TTY, printers, CRT s or microprocessors. Generates 14 different standard bit rates which are multiplied under external control to 1X, 8X, 16X or 64X initial value. Operates from single +5 volt supply. MC14411........................ $11.98
4 pages of data.................................. .40
Crystal for the above........................... $4.95

- Accuracy: ±0.05% of Reading ±1 Count
- Two Voltage Ranges: 1.999 V and 199.9 mV
- Up to 25 Conversions/s
- $Z_{in} > 1000$ M ohm
- Auto-Polarity and Auto-Zero
- Single Positive Voltage Reference
- Standard B-Series CMOS Outputs - Drives One Low Power Schottky Load
- Uses On-Chip System Clock, or External Clock
- Low Power Consumption: 8.0 mW typical @ ±5.0 V
- Wide Supply Range: e.g., ±4.5 V to ±8.0 V

MC14433 SINGLE CHIP 3½ DIGIT A/D

Single chip combines linear and CMOS digital to bring you the simplest yet DVM approach. Requiring only 4 external passive parts, this subsystem gives you: Auto polarity, auto zero, single voltage reference, 8 mW operation, overrange, underrange signals, 25 conversions per second and .05% ± 1 count accuracy! 100 uV resolution. 24 Pin DIP.
MC14433P...........with specs................$19.55

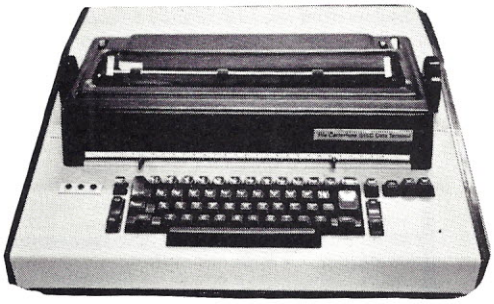
tri-tek, inc.
6522 north 43rd avenue,
glendale, arizona 85301
phone 602 - 931-6949

We pay surface shipping on all orders over $10 US, $15 foreign in US funds. Please add extra for first class or air mail. Excess will be refunded. Orders under $10, add $1 handling. Please add 50¢ insurance. Master charge and Bank America cards welcome, ($20 minimum). Telephone orders may be placed 10AM to 5:30PM daily, Mon thru Fri. Call 602-931-4528. Check reader service card or send stamp for our latest flyers packed with new and surplus electronic components.

RONDURE COMPANY

2522 BUTLER ST. • DALLAS, TEXAS 75235 • 214-630-4621

the computer room

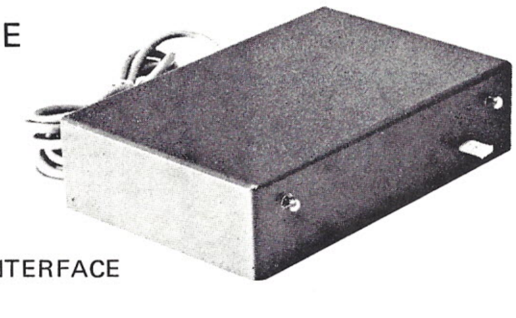


Specifications
- Size: 21″ wide x 21″ deep x 8″ high
- Power Input 115 Volt 60 Hz
- Interface: RS232
- Weight: 54 lbs. (Shipping Weight 65 lbs.)
- 15″ Carriage
- Input/Output rates to 15 characters per second
- EBCD Code
- Half Duplex
- 132 Print Positions, 10 Pitch
- Can be used off-line

Used
Working
(Non Refurbed) $695.
Used
Working
(Refurbed) $895.
Software to connect ASCII Output of 8080 Class Processor to Selectric: Code $25
Manufacturers Electronic & Mechanical Documentation
$20. with machine $40. Documentation only

SELECTRIC TERMINAL (IBM Selectric Mechanism, Heavy Duty, Datel Electronics)

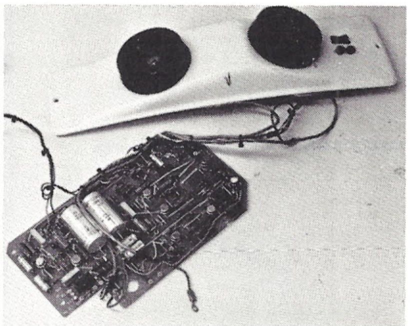
CARTERFONE MODEL 318 ASYNCH MODEM



- HARD WIRE
- TTY OR RS-232B INTERFACE
- ORIGINATE ONLY
- UP TO 300 BPS

USED – UNTESTED . . $25.00
USED – TESTED $80.00

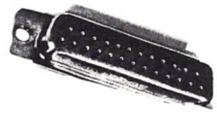
We ship prints with these.



ACCOUSTICAL MODEMS – ORIGINATE ONLY USED – UNTESTED

Physically fit into Model 33 Teletype. Manufactured by **Paragon** partial documentation 2 for $25

CANNON 25 PIN CONNECTOR



ITEM C-1
RS232
Male Connector *NEW* **$2.50** Each
Solder Type Cover **$1.00** Each

SHIPPING INFORMATION:

Modems: $2.00 each; 2 for $4.00 UPS
Small Items & Parts: $2.00/order less than $20.00; $4.00/order $20.00 to $100.00; $6.00/order over $100.00
Large Items & Parts: Specify Freight or Air Freight Collect
Foreign Orders: Add appropriate freight or postage
Please specify exactly what you wish by order number or name or both.
We now take Master Charge orders. Specify full number, bank number and expiration date.

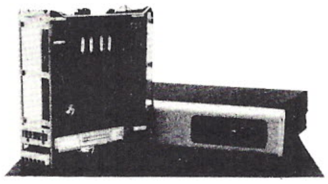
ORDERING INFORMATION:

All items subject to availability. Your money returned if we are out of stock.
Items are either new (specified) or they are used (tested or untested) and no other warranty is made or implied.

DEC LSI-11 COMPONENTS

All Items NEW

ORDER BY PART NUMBER



		List	Selling Price
CPU	(KDIIF)	$990	$875
SIO	(DLVII)	235	210
Card Cage	(H9270)	175	155
Ext. Arith.	(KEVII)	175	155
PIO	(DRVII)	195	175
4K RAM (8KB)	(MSVIIB)	625	550
PROM/ROM Bond	(MRVIIAA)	175	155

Write for our CATALOG of many parts terminals printers etc

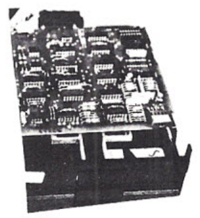
In general no cords or cables are shipped unless we specify that they are supplied.
We ship the same day we receive a certified check or money order.
Texas residents add 5% sales tax.
Please call if you have a question.

SUGART MINI-FLOPPY DRIVE

NEW

$390.00 each

MODEL SA-400

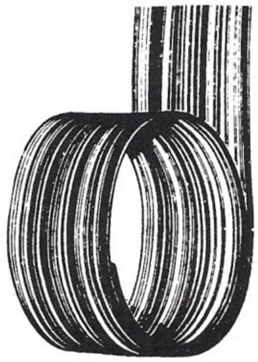


RONDURE COMPANY R7
2522 BUTLER
DALLAS, TEXAS 75235
PHONE: (214) 630-4621

SPECTRA FLAT TWIST

50 conductor, 28 gauge, 7 strands/conductor made by Spectra. Two conductors are paired & twisted and the flat ribbon made up of 25 pairs to give total of 50 conductor. May be peeled off in pairs if desired. Made twisted to cut down on "cross talk." Ideal for sandwiching PC boards allowing flexibility and working on both sides of the boards. Cost originally $13.00/ft

SP-324-A $1.00/ft. 10 ft/$9.00



SP-234-A	$1.00 ft	50 cond.	10 ft/$9.00
SP-234-B	.90 ft	32 cond.	10 ft/$8.00

WIRE WRAP WIRE

TEFZEL blue #30 Reg. price $13.28/100 ft. Our price 100 ft $2.00; 500 ft $7.50.

MULTI COLORED SPECTRA WIRE

Footage		10'	50'	100'
8 Cond.	#24	$2.50	9.00	15.00
12 "	22	3.00	11.00	18.00
14 "	22	3.50	13.00	21.00
24 "	24	5.00	20.00	30.00
29 "	22	7.50	28.00	45.00

Great savings as these are about 1/4 book prices. All fresh & new.

TOUCHTONE ENCODER CHIP

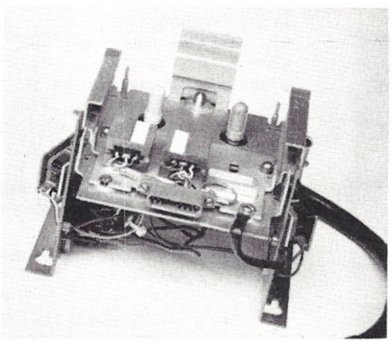
Compatible with Bell system, no crystal required. Ideal for repeaters & w/specs. $6.00

CHARACTER GENERATOR CHIP

Memory is 512x5 produces 64 five by seven ASCII characters. New material w/data. $6.00

FIRE-SMOKE-INTRUDER ALARM

12 volt DC operation, large 5 inch bell, one unit, all the works under the bell. Nice for camper, boat, homes. $25



VIATRON CASSETTE DECKS

The computer cassette deck alone. $35
Set of 2 boards read/write amp & servo control boards for this deck. $40.00

PORTABLE FIRE ALARM

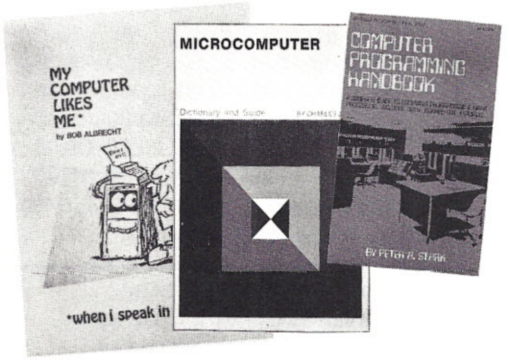
Operates on 3 internal AA cells (not furnished) On temp rise of approx 150° Sonalert type alarm sounds off. Push to test switch incorporated. The push to test switch makes nice sounding code practice oscillator. $6.00

Meshna

Please add shipping cost on above. Minimum order $10
FREE CATALOG SP-9 NOW READY
P.O. Box 62, E. Lynn, Massachusetts 01904

M2

KB BOOK NOOK



●BRAND NEW DICTIONARY This new microcomputer dictionary fills the urgent need for all computer people, engineers, scientists, industrialists, communications people — as professionals, amateurs, teachers, or students — to become quickly acquainted with the terminology and nomenclature of a new revolution in computer control capabilities in areas that pervade most of man's daily activities.

Over 5000 definitions and explanations of terms and concepts (704 pages) relating to microprocessors, microcomputers and microcontrollers. There are also separate appendices on: programmable calculators; math and statistics definitions; flowchart symbols and techniques; binary number systems and switching theory; symbol charts and tables; summaries of BASIC FORTRAN and APL. In addition there is a comprehensive electronics/computer abbreviations and acronyms section. $15.95.

●COMPUTER PROGRAMMING HANDBOOK by Peter Stark. A complete guide to computer programming and data processing. Includes many worked out examples and history of computers. $8.95

●MY COMPUTER LIKES ME . . . WHEN I SPEAK BASIC An introduction to BASIC . . . simple enough for your kids. If you want to teach BASIC to anyone quickly, this booklet is the way to go. $2.00.

●COMPUTER DICTIONARY by Donald D. Spencer. A compact compendium of computer terms for beginners and professionals alike. Defines words and acronyms used by computerists in a clear, easy to understand style. Over 2000 definitions are provided. This reference is a must for the individual getting started in the world of microcomputers. Published by Camelot Press, $5.95.

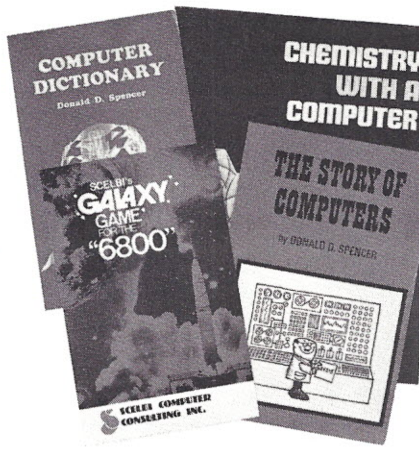
●SCELBI'S GALAXY GAME FOR THE "6800" Here's a new twist in computer games by Scelbi Computer Consulting and Robert Findley/Raymond Edwards. The game, "Galaxy" pits the operator of a spaceship against alien craft, as well as such variables as speed, time, and ammunition. No two games are the same! This game is described in *Galaxy Game for the 6800,* published by Scelbi Computer Consulting, Inc. $14.95

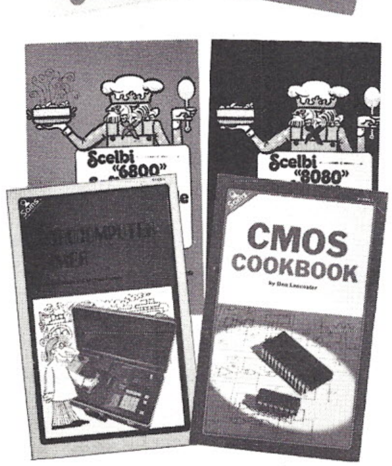
●6800 SOFTWARE GOURMET GUIDE & COOKBOOK If you have been spending too much time developing routines for your 6800 microprocessor, try the new book by Scelbi Computing and Robert Findley. This manual, *6800 Software Gourmet Guide and Cookbook* described sorting, searching, and many other necessary routines for the 6800 user. $9.95.

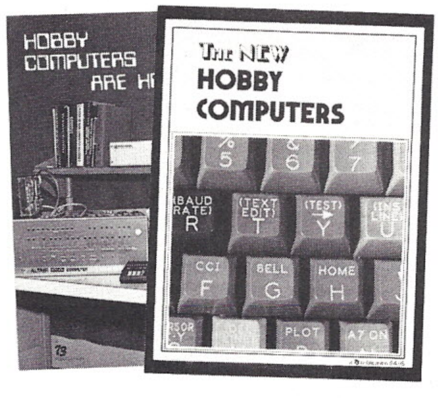
●8080 SOFTWARE GOURMET GUIDE AND COOKBOOK If you have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. This manual, *8080 Software Gourmet Guide and Cookbook* described sorting, searching, and many other routines for the 8080 user. $9.95

●CMOS COOKBOOK by Don Lancaster, pub. Howard W. Sams Company. Another winner from Don Lancaster, author of the famous *RTL* and *TTL Cookbooks.* The *CMOS Cookbook* details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. The book follows the style of the original Cookbooks. Eight chapters cover all facets of CMOS logic, and the work is prefaced by 100 pages detailing the characteristics of most CMOS circuits. The *CMOS Cookbook* is required reading for every serious digital experimenter. $9.95

●HOBBY COMPUTERS ARE HERE If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, the basics of programming, along with a couple TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95.







●CHEMISTRY WITH A COMPUTER by Paul A. Cauchon. A chemistry book which contains a collection of tutorial, simulation and problem-generation computer programs. Tutorials provide individualization of assignment, immediate evaluation of responses and a new set of problems with each run. Simulations provide models of lengthy laboratory experimentation beyond the limited classroom timeframe and enhancement of course studies by encouraging prelaboratory research. Problem-generating programs provide individualized sets of questions on a given topic. Can be used with almost any chemistry course at the high school or college level. $9.95.

●THE STORY OF COMPUTERS by Donald D. Spencer is to computer books what *Dick and Jane* to novels . . . extremely elementary, gives the non-computerist a fair idea of what the hobbyist is talking about when he speaks computer lingo. Attempts to explain what computers are and can do to a spouse, child or any un-electronics-minded friend. $4.95.

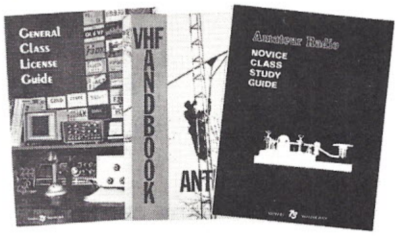
●MICROCOMPUTER PRIMER by Mitchell Waite and Michael Pardee, pub. by Howard W. Sams Company. If you are afraid to get involved with microcomputers for fear of not understanding them, worry no longer! The *MICROCOMPUTER PRIMER* describes basic computer theory, explains numbering systems, and introduces the reader to the world of programming. This book does not elaborate on specific systems or chips, but describes the world of microcomputing in "real world" terminology. There is probably no better way of getting involved with the exciting new hobby of microcomputing. $7.95

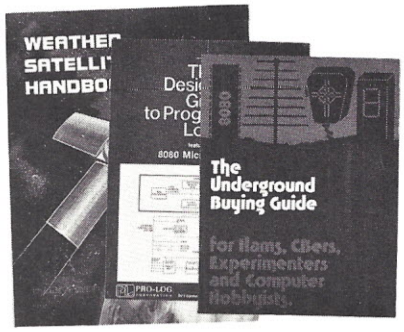
● INTRODUCTION TO MICROPROCESSORS by Charles Rockwell of MICROLOG Here is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. This book describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Addressing schemes, registers, control, and memory are all explained, and general information about hardware systems is provided. Specific systems are not described and programming is only briefly discussed. *Introduction To Microprocessors* is a hardware introduction . . . and a good one. $17.50 US and Canada, $20 elsewhere.

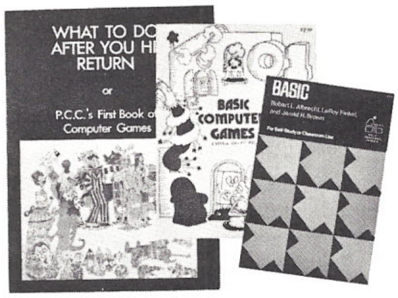
●THE NEW HOBBY COMPUTERS! This book takes it from where "Hobby Computers Are Here" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! Everything of interest is there in one volume, ready to be enjoyed by you. $4.95.

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud Book Dept., Peterborough NH 03458. Be sure to include check or detailed credit card information.
Note: Prices subject to change on books not published by 73 Magazine.

KB BOOK NOOK







●**NOVICE STUDY GUIDE** This is the most complete Novice study guide available. It is brand new. This is not only invaluable for anyone wanting to get started in amateur radio, but also it is about the only really simple book on the fundamentals of electricity and electronics. And without your fundamentals down pat, how can you go on to really understand and work with computers? First things first. $4.95

●**GENERAL CLASS STUDY GUIDE** This book takes over on theory where the Novice book leaves off. You'll need to know the electronic theory in this to work with computers and you'll not find an easier place to get the information. It will also make getting your Tech or General license a breeze . . . then you can get on the ham repeaters and interconnect your micro with others. $5.95

●**VHF ANTENNA HANDBOOK** The NEW VHF Antenna Handbook details the theory, design and construction of hundreds of different VHF and UHF antennas . . . a practical book written for the average amateur who takes joy in building, not full of complex formulas for the design engineer. Packed with fabulous antenna projects you can build. $4.95

●**WEATHER SATELLITE HANDBOOK** Simple equipment and methods for getting good pictures from the weather satellite. Antennas, receivers, monitors, facsimile you can build, tracking, automatic control (you don't even have to be home). Dr. Taggart WB8DQT $4.95

●**SSTV HANDBOOK** This excellent book tells all about it, from its history and basics to the present state of the art techniques. Contains chapters on circuits, monitors, cameras, color SSTV, test equipment and much more. Hardbound $7, Softbound $5

●**WHAT TO DO AFTER YOU HIT RETURN** PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, muchly illustrated. Lunar landing, Hammurabi, King, Civil 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $8.00

●**101 GAMES IN BASIC** Okay so once you get your computer up and running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you, from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50

●**BASIC** by Bob Albrecht, etc. Self-teaching guide to the computer language you will need to know for use with your microcomputer. 324 pages. This is one of the easiest ways to learn computer programming. $4.95

●**TVT COOKBOOK** by Donald Lancaster, describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. This book is a required text for both the microcomputer enthusiast and the amateur RTTY operator who desires a quiet alternative to noisy teletype machines. $9.95

●**TTL COOKBOOK** by Donald Lancaster. Explains what TTL is how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter, and a digital tachometer. 336 pages; 5½ x 8½; softbound $8.95

●**AN INTRODUCTION TO MICROCOMPUTERS, VOLS. 1 AND 2** by Adam Osborne Associates, are references dealing with microcomputer architecture in general and specifically with details about most of the common chips. These books are not software-oriented, but are invaluable for the hobbyist who is into building his own interfaces and processors. Volume 1 is dedicated to general hardware theory related to micros, and Volume 2 discusses the practical details of each micro chip. (Detailed review in K'lobaud #2) $7.50 each

●**8080 PROGRAMMING FOR LOGIC DESIGN** Here is an ideal reference for the person desiring an in-depth understanding of the 8080 processor. The work is application-oriented, and the 8080 is discussed in light of replacing conventional, hard-wired logic systems. Both hardware and software is described. Practical design considerations are provided for the individual wishing to implement an 8080-based control system. (Detailed review in Kilobaud #1) Published by Osborne Associates, $7.50.

●**6800 PROGRAMMING FOR LOGIC DESIGN** Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Both hardware and software techniques are discussed, as well as interface information. This reference, and its companion dedicated to 8080 users, provide practical information that allows an experimenter to design a complete micro control system from the "ground up." An excellent reference! Published by Osborne Associates, $7.50.

●**THE UNDERGROUND BUYING GUIDE** Here is a handy guide for the electronics enthusiast. Over 600 sources of equipment and literature are provided; some are mail-order-only outfits that do not advertise. Sources are listed alphabetically, by service or product, and by state. The guide is cross-referenced for ease of use. Electronic publishing houses are also listed. Published by PMS Publishing Co., $5.95 each.

Test Equipment Library

●**VOL. I COMPONENT TESTERS** Build your own test equipment and save a bundle (and have a lot of fun). Volume I of the 73 Test Equipment Library shows you how to build and use transistor testers (8 of 'em), three diodes testers, 3 IC testers, 9 voltmeters and VTVMs, 8 ohmmeter, 3 inductance meters, and a raft of other gadgets for checking temperature, crystals, Q, etc. $4.95

●**VOL II AUDIO FREQUENCY TESTERS** If you're into audio . . . such as digital cassette recording, RTTY, Baudot vs ASCII, SSTV, SSB, Touchtone or even hi-fi . . . you'll want to have this book full of home built test equipment projects. Volume II $4.95

●**VOL. III RADIO FREQUENCY TESTERS** This is of more interest to hams and CBers . . . test equipment you can build for checking out transmitters and receivers: signal generators, noise generators, crystal calibrators, GDOs, dummy loads . . . things like that. This is Volume III of the 73 Test Equipment Library $4.95

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud Book Dept., Peterborough NH 03458. Be sure to include check or detailed credit card information.
Note: Prices subject to change on books not published by 73 Magazine.

KB CALENDAR

New Haven

The Southern New England Computer Society meets the third Sunday of each month not far from I-91. Meetings have been held as far south as New Haven and as far north as Windsor Locks. Call Charlie at (203) 562-4739 or write SNECS, 267 Willow St., New Haven, CT 06511.

Computer Show Set for October

The world's largest Holiday Inn at Chicago O'Hare International Airport is the setting for the next Personal Computing Show to be held October 27, 28, 29.

With space for over 100 exhibits, the show will feature a variety of personal computer systems, new products, home brewed systems and applications – all of it directed at the computer neophyte. Manufacturers and distributors will be offering consumer discounts for cash purchases at the show – some up to 50%! Door prizes, grand prizes, gifts and surprises of special interest to computer hobbyists and amateurs will all be the order of this Personal Computing Show.

Computer enthusiasts are encouraged to participate in the show. Since the primary purpose is to explain all aspects of personal computing to the public, there will be a need for dozens of workshops and seminars. Plans call for publishing all papers, to be made available after the show is over.

If you are interested in participating in this show, contact David Bunnell or Louise Garcia (505) 266-1173, no later than August 15, 1977.

San Jose

The fourth annual conference on Computer Graphics and Interactive Techniques, sponsored by SIGGRAPH (ACM's special interest group on graphics), will be held in San Jose, California, at the Hyatt House Hotel from July 18-22, 1977. SIGGRAPH Week begins with two comprehensive 2-day workshops (July 18-19). The vendor exposition will be the largest specialized computer graphics tool display ever, and will be open to the general public for a $5.00 admission. For further information contact: Stephen Levine, Lawrence Livermore Laboratory, Box 808 MS L-73, Livermore CA 94550.

MIMI '77 Montreal

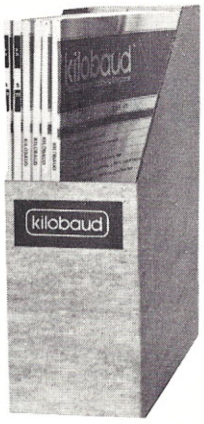
A call for papers has been put out in preparation for the MIMI '77 Montreal International Symposia. A 200-250 word abstract should be submitted by September 1, 1977. The Symposia is scheduled for November 16-18 at the Queen Elizabeth Hotel in Montreal. Send abstracts or requests for information to Prof. K. L. Houle – MIMI '77, Ecole Polytechnique, Case Postale 6079, Succursale A, Montreal Quebec Canada H3C 3A7.

Houston

September 16-18, Houston Personal Computing Faire; contact Richard McClendon, PO Box 36584, Houston TX 77036.

what do you give the man who has everything?

a box to put it in.



Kilobaud, as thick as it is, is more like a floppy when it comes to standing on the bookshelf. Enter the new Kilobaud Library Shelf Boxes, sturdy corrugated cardboard boxes which will hold your magazines on the shelf and keep them from flopping around.

Yes, we know all about binders . . . we have them too . . . and we sell them, but binders are a drag when you want one copy of a magazine. And they cost like sin (which costs plenty).

Just to be rotten (a talent we are trying to develop, but which comes hard), we have self-sticking labels for the boxes, not only for *Kilobaud*, but also for *73 Magazine* . . . and for *Personal Computing, Radio Electronics, Popular Electronics, Interface Age*, and . . . yep . . . *Byte*. Heh, heh! Just ask for whatever stickers you want and we'll throw 'em in with your box order. Hams may want our labels for *CQ, QST* or *Ham Radio*, if they get any of those magazines. This is a way you can buy one set of matching boxes and line 'em up on your shelf . . . looks very nice that way.

The boxes are a white color and are particularly resistant to dirt, a real plus for white boxes. There's some kind of funny plastic finish on 'em.

You'll probably do like most people who have tried these so far and order one or two for starters . . . then get a couple dozen. The postage on these is the killer . . . so one box costs $2.00 postpaid and ten or more are $1.50 each postpaid.

Unless the magazine gets whole lot fatter than it is right now these boxes should hold a full year of *Kilobaud* . . . or 73.

One side is cut low to permit you to see the binding of the magazine . . . and note that we are now printing the information on the top part of the binding so it will show in these boxes. You can put the boxes on your shelves with the bindings showing or with just the white board showing, there are little marks to help you center your labels on either side.

Your magazine library is your prime reference, so keep it handy and keep it neat with these strong library shelf boxes.

Send me ______ boxes for Kilobaud Shelf Storage at $2.00 for the first box and $1.50 for each additional box. Include the following labels: ______________

$_____ Enclosed ☐ Cash ☐ Check ☐ M.O.

Bill: ☐ American Express ☐ BankAmericard ☐ Master Charge – Interbank #_______

Card #______________ Expiration date______

Signature__________________

Name__________________

Address__________________

City__________ State ______ Zip______

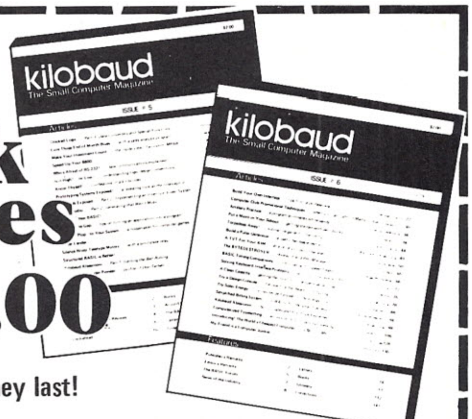
KILOBAUD LIBRARY SHELF BOXES
Peterborough NH 03458
or call Toll Free (800) 258-5473
NH & Evenings (800) 251-6771

KB/8/77

Compatible Tone Dialer

The "Soft-Touch" Tone Dialer from TEC is compatible with all Western Electric telephone handsets. The Tone Dialer gives the user instant tone capability from any standard rotary dial telephone. Soft-Touch uses LSI circuitry equivalent to 4,000 transistors. Electret microphone gives 0.1% distortion for error-free modem use. Unique design allows you to softly touch the face panel to generate the exact tones used to place telephone calls or to input or control equipment remotely. BAC, YISA, & MC accepted. Priced at $29.95 from Telephone Electronics Corp., 615 Third St., San Francisco CA 94107.

KB back issues $3.00



While they last!

Did you manage to miss out on the first issues of Kilobaud? Don't chance not getting these action packed thrillers. While they last they are available for the astounding (we have *a lot* of gall) price of only *(only?)* $3.00 each postpaid (and that's a big deal, with each copy running us 72¢ postage). Domestic orders only.

Please send me KILOBAUD Back Issues at $3 each!

___ issues JAN 77 ___ issues APR 77 ___ issues JUL 77
___ issues FEB 77 ___ issues MAY 77 ___ TOTAL 8/77
___ issues MAR 77 ___ issues JUNE 77

☐ BankAmericard ☐ Master Charge ☐ American Express

Card #______________ Interbank #______

Expiration date ______

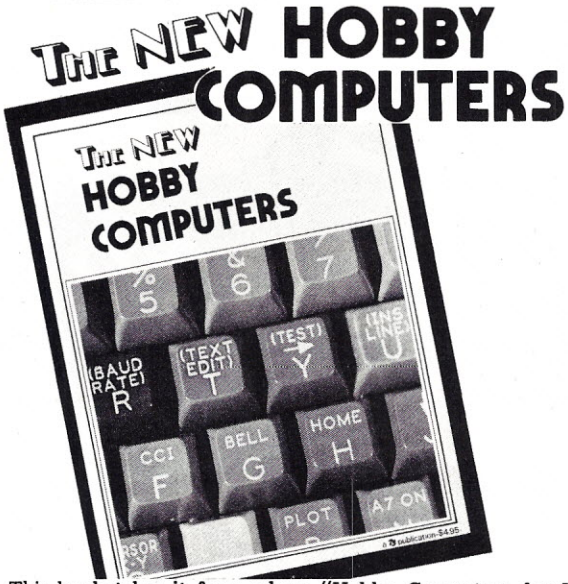
Signature ______________

Name ______________

Address ______________

City __________ State ______ Zip ______

Have credit card #handy and call
TOLL FREE (800) 258-5473 NH & Nights (800) 251-6771
KILOBAUD • PETERBOROUGH NH 03458



This book takes it from where "Hobby Computers Are Here" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! Everything of interest is there in one volume, ready to be enjoyed by you. Don't miss this tremendous value! Only $4.95

HOBBY COMPUTERS ARE HERE

"It's the **first book I've ever read about computers** that I can understand . . ."

HOBBY COMPUTERS ARE HERE helps the beginner get into the world of microcomputers. Some chapters: What's a Computer?, Is Digital New?, How Computers Figure, Computer Languages, How Gates Work, TTL – Best Logic Yet, Ins and Outs of TTL, Flip-Flops Exposed, New Cassette System Standard, Build this TVT, Using Surplus Keyboards, Morse to RTTY Converter, ASCII to Baudot via a PROM, PLUS reprints of some of the 73 editorials on computers. Don't miss out any longer on the fun of hobby computing and the fantastic applications of these incredible devices! $4.95

73 magazine Dept. HC
Peterborough NH 03458

For ease in ordering, please use order card in the back of this magazine.

KB 8/77

BEST Computer Mailing List

By far the most complete mailing list available is the KILOBAUD list of DEALERS, CLUBS, PUBLICATIONS and MANUFACTURERS. (It's the one we use for our mailings and we update it daily). The list has over 1000 names painstakingly gathered from manufacturers, magazine ads and new product releases, hobby computer shows and direct mail. You can buy this list printed on self-sticking labels for only $50.

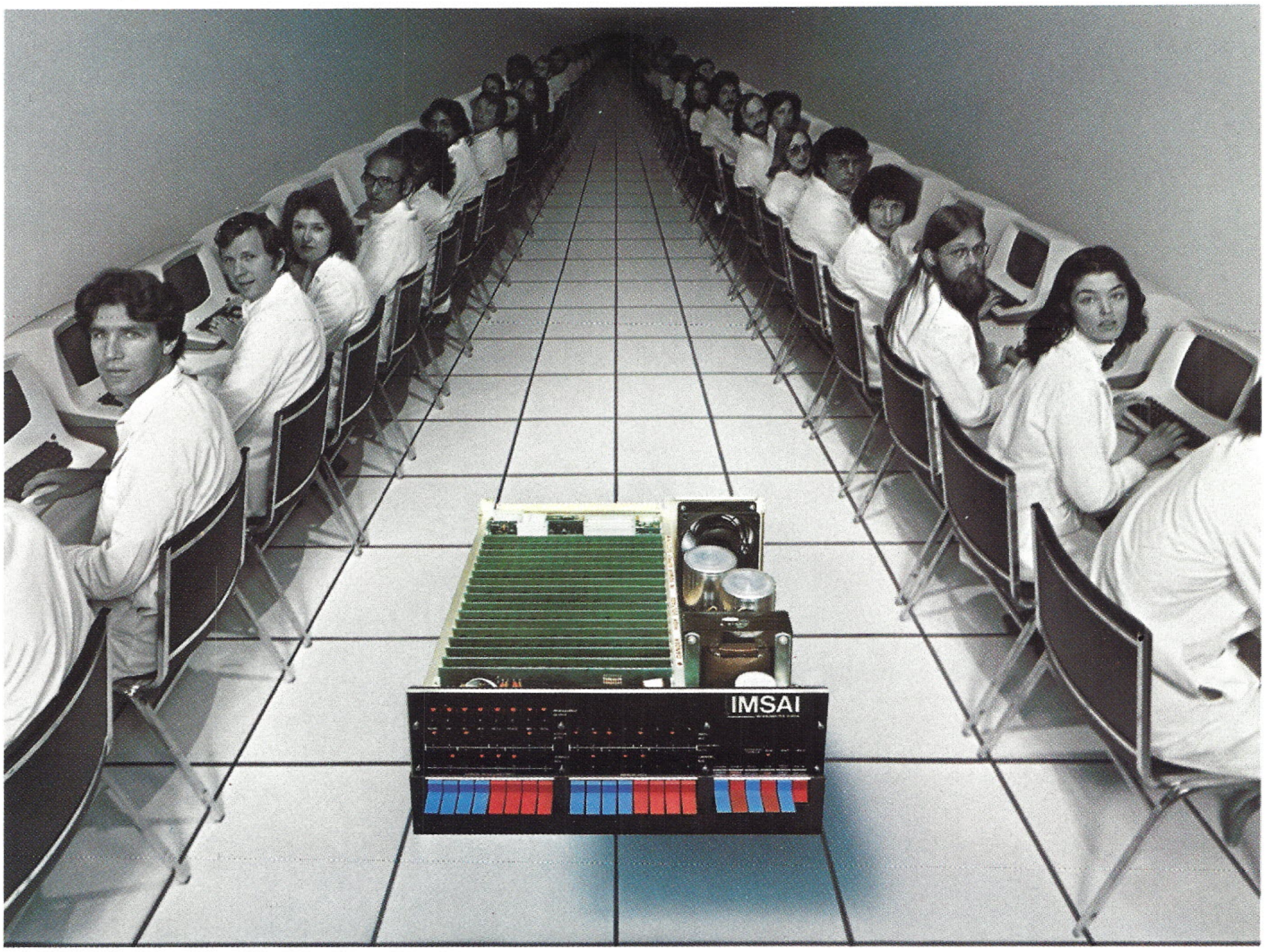
Additional printouts, once you are a customer, are $35. Call in your order with charge information (BAC, AMEX, MC). Our toll free number for these orders is (800) 258-5473.

NEW FIRMS, DEALERS, CLUBS . . . be sure we have your name, address, phone number and as much data as possible for this listing.

kilobaud PETERBOROUGH NH 03458

KB/8/77

POWER.



IMSAI Introduces the Megabyte Micro.™

The Megabyte Memory

Until today, the largest memory you could fit and address in a single microcomputer CPU was 65K.

Now, IMSAI presents an incredible memory system for micros 16 times more powerful than yesterday's best.

Imagine, a full megabyte of power from sixteen 65K RAM boards.

And, to control all this, the IMSAI Intelligent Memory Manager (IMM), the super control board.

You can write protect blocks throughout the full megabyte. Or, map in 16K blocks.

Plus, preset 16 mapping configurations with protect for high speed transfer or rapid change.

All interrupts are fully vectored, and there's an interrupt if an attempt is made to write into protected memory.

There's even a real "time of day" clock.

65K, 32K and 16K RAM Boards

Until today, the most memory you could plug into a single slot was 16K.

Now, IMSAI presents memory boards in astonishing multiples of sixteen: 65K, 32K and 16K low power, dynamic RAM Boards. They can be used in any S-100 bus computer individually or in combination to form conventional systems up to 65K bytes.

Every board is fast. With "hidden refresh" and *no* "wait state."

The Complete Megabyte Microcomputer System

The IMSAI Megabyte Micro™ is only part of the story. The full system can include dual floppy disks, terminals, plotters, printers and tape cassettes.

IMSAI also offers the finest high level and peripheral software available. Paper tape and Tape Cassette I/O and super Disk Operating Systems. Plus, BASIC and Disk BASIC with more high level languages coming.

Until today, the microcomputer's potential was just something you talked about.

Now, you can put it to work. Powerfully.

GENTLEMEN:
I'm power hungry!
☐ Send 65K RAM Board Kit $2599 ☐ Assembled $3899
☐ Send 32K RAM Board Kit $749 ☐ Assembled $1099
☐ Send 16K RAM Board Kit $449 ☐ Assembled $679
☐ Send IMM ROM Control Kit $299 ☐ Assembled $399
☐ Send IMM EROM Control Kit $499 ☐ Assembled $699
☐ Send full catalog $1.00

Check/MO enclosed. Amt. $ ________
Charge my: ☐ BAC ☐ M/C

________ Exp. Date: ________

Sig. ________

☐ Send name of my nearest IMSAI dealer

Name ________

Company ________ Title ________

Address ________

City ________

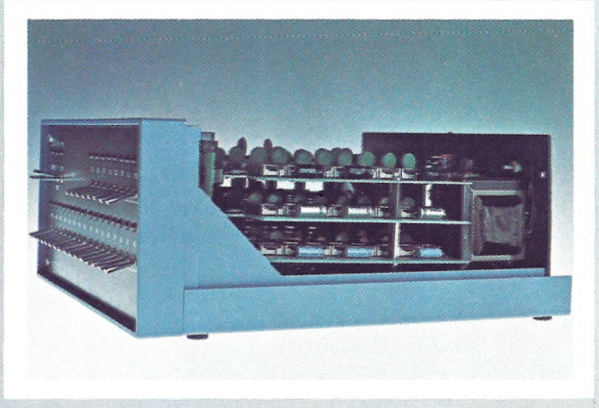
State/Zip ________

IMSAI®

IMSAI Manufacturing Corporation
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093 TWX 910-366-7287

Powerful in computing muscle, yet small in physical size, the Altair™680b offers many special features at an affordable price. Based on the 6800 microprocessor, the 680b comes with 1K of static RAM, Serial I/O port, PROM monitor and provisions for 1K of PROM as standard components. It's good thinking, when you're interested in making a modest investment on a highly reliable computer, to consider the Altair 680b.

Our PROM monitor eliminates the necessity for toggling front panel switches to load bootstraps or manipulate memory contents. Only a terminal and programming language are required for complete system operation. With Altair System software—Altair 680 BASIC, assembler and text editor—you may begin problem solving immediately with ease.



By adding the 680b-MB Expander card, many options are currently available:

*16K Static Memory Board—Increase your system memory with 16K bytes of fast access (215 ns), low power (5 watts per board) static RAM. 680 BASIC and assembler/text editor are included free with purchase.

*Process Control Interface—A PC card that uses optically isolated inputs and relay outputs that transmit sensory information to and control signals from the computer. A diverse world of control applications is opened up with the Altair 680b-PCI.

*Universal Input/Output Board—If your I/O needs exceed the serial port already on the main board, augment your I/O channels with the 680b-UI/O. By implementing the optional serial port and two parallel ports, you can simultaneously interface to four terminals.

*New Addition—Kansas City Audio Cassette Interface—Use the 680b-KCACR to interface your Altair 680b with an audio cassette recorder for inexpensive mass storage of programming languages, programs and data.

Available in either full front panel or turnkey models, the Altair 680b presents many computing capabilities at a low cost—without skimping on performance. See it today at your local Altair Computer Center or contact the factory for further details.

mits

Good Thinking.

2450 Alamo S.E. Albuquerque, New Mexico 87106
dealer inquiries invited.